

Classroom Game

La Universidad Central de Asia (CAU) celebra su aniversario organizando un juego entre estudiantes y profesores. En el salón de clases hay N estudiantes, numerados de 0 a $N - 1$. Cada estudiante recibe una hoja de papel que contiene un arreglo de enteros. Inicialmente, cada hoja está vacía, por lo que contiene un arreglo vacío.

Los estudiantes juegan con M profesores, numerados de 0 a $M - 1$, y el director. El juego consiste de M rondas, numeradas igualmente de 0 a $M - 1$. En la ronda j , el profesor j entra al salón y ocurre lo siguiente:

- Algunos estudiantes (o ninguno) levantan la mano. Se garantiza que en cada ronda al menos un estudiante **no** levanta la mano, y cada estudiante levanta la mano **a lo más una vez** durante el juego completo.
- El profesor observa el estado actual de las hojas de los estudiantes, y decide si modificar o no las hojas de los estudiantes que **no** levantaron la mano en la ronda actual. Para cada estudiante, el profesor puede reemplazar el contenido de su hoja con un nuevo arreglo (puede estar vacío) de enteros. Cada entero deberá estar entre 0 y 63 inclusive, y los arreglos pueden tener a lo más 63 elementos.
- Después de estas modificaciones, el profesor j se va y los estudiantes intercambian sus hojas de acuerdo a una permutación oculta P . Siguiendo esta regla, cada estudiante i ($0 \leq i < N$) le entrega su hoja al estudiante $P[i]$, donde $P[0], P[1], \dots, P[N - 1]$ son enteros **distintos** entre 0 y $N - 1$, inclusive. Los valores de P están **fijos** durante todas las rondas del juego, pero ni los profesores ni el director los conocen.

Una vez concluidas las M rondas y se haya realizado el último cambio indicado por la permutación P , el director entra al salón. Con solo mirar a las hojas que sostienen los estudiantes, el director debe determinar, para cada estudiante i ($0 \leq i < N$), la ronda número j en la que el estudiante i levantó la mano, o indicar que el estudiante i nunca levantó la mano.

Los profesores no pueden comunicarse entre ellos o con el director, excepto a través de los arreglos de enteros escritos en las hojas de los estudiantes.

Tu tarea es diseñar e implementar la estrategia para los profesores y el director de manera que pueda identificar correctamente para cada estudiante si levantó la mano y en caso de hacerlo cuándo lo hizo. Tu puntaje dependerá de la máxima longitud de cualquier arreglo escrito en alguna hoja, entre más pequeño sea este valor mayor será tu puntaje.

Detalles de implementación

Deberás implementar dos funciones, una para los profesores y otra para el director.

La función que debes implementar para los profesores es:

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

- N : el número de estudiantes.
- M : el número de rondas y a su vez el número de profesores.
- R : la ronda actual (de 0 a $M - 1$).
- T : un arreglo (puede estar vacío) indicando a los estudiantes que levantaron la mano en esta ronda, ordenados de manera creciente.
- A : un arreglo de longitud N describiendo el contenido de las hojas, donde $A[i]$ ($0 \leq i < N$) es el arreglo de enteros en la hoja que sostiene el estudiante i al inicio de la ronda.
- Esta función se llama exactamente M veces por juego en el orden $R = 0, 1, \dots, M - 1$.

Esta función debe retornar un arreglo B de longitud N , representando las hojas después de las modificaciones del profesor, en el mismo formato que el arreglo A .

- Para cada estudiante i que levantó la mano (en otras palabras, i aparece en T), su hoja no debe ser modificada.
- Para cada i tal que $0 \leq i < N$, la longitud de $B[i]$ no debe superar 63, y cada entero en $B[i]$ debe estar entre 0 y 63, inclusive.

La función que debes implementar para el director es:

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

- N, M : definidos previamente.
- A : un arreglo de longitud N , donde $a[i]$ es el arreglo de enteros escrito en la hoja que sostiene el estudiante i después de las M rondas.
- Esta función se llama exactamente una vez por juego, después de la última llamada a `process_step`.

Esta función debe retornar un arreglo D de longitud N . Para cada i tal que $0 \leq i < N$, se debe cumplir que

- $D[i] = j$ si el estudiante i levantó la mano durante la ronda j , o
- $D[i] = -1$ si el estudiante i nunca levantó la mano durante todo el juego.

Tu programa no deberá guardar información o comunicar información entre las llamadas a `process_step`, a excepción del arreglo retornado por la función `process_step`. Si tu programa intenta hacer esto, tu puntaje en el CMS podrá no ser correcto y podrá ser reducido al final del concurso. Además, el evaluador podrá realizar múltiples ejecuciones de tu programa simultáneas, Esto significa que las llamadas a `process_step` y `determine_steps` podrán ser llamadas en ejecuciones distintas de tu programa, y que llamadas a `process_step` y `determine_steps` de diferentes juegos podrán ser ejecutados en la misma instancia. Cada caso incluye hasta 5 juegos.

Límites

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Cada estudiante levanta la mano **a lo más una vez** durante el juego completo. En otras palabras, para cada estudiante i , existe a lo más una ronda j en la cual el estudiante i levanta la mano.
- En cada paso, al menos un estudiante **no** levanta la mano..

Subtareas y ponderación

Subtarea	Puntaje	Restricciones adicionales
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ para cada $0 \leq i \leq N - 1$.
4	25	A lo más un estudiante levanta la mano en cada ronda.
5	56	Sin restricciones adicionales.

Para cada caso, tu puntaje es 0 (reportado como `Output isn't correct` en CMS) si el valor de retorno de alguna llamada a la función `process_step` es inválida o si el valor de retorno de cualquier llamada `determine_steps` es incorrecto.

En caso contrario, sea C la máxima longitud de **algún** arreglo en **algún** arreglo B retornado por `process_step`. Tu puntaje para un caso en una subtarea con valor S se calcula como $S \cdot X$, donde X depende de C de acuerdo a la tabla siguiente:

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

Ejemplo

Considere el escenario con $N = 4$ estudiantes, $M = 2$ profesores, y la permutación $P = [0, 3, 1, 2]$. Inicialmente, todas las hojas están vacías.

El evaluador llama primero a :

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

El estudiante 1 levantó la mano, por lo que su hoja no puede ser modificada. El profesor 0 decide escribir $[10]$ en la hoja del estudiante 0, escribir $[1, 63, 4]$ en la hoja del estudiante 3, y dejar la hoja del estudiante 2 vacía. Para esto, la función debe retornar $B = [[10], [], [], [1, 63, 4]]$.

Luego el profesor 0 se va y los estudiantes intercambian sus hojas de acuerdo a la permutación P . Después del intercambio, las hojas son $A = [[10], [], [1, 63, 4], []]$

El evaluador ahora llama a:

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

Los estudiantes 0 y 3 levantaron la mano, así que sus hojas no serán modificadas. El profesor 1 decide escribir $[0, 40]$ en la hoja del estudiante 1 y escribir $[1, 50]$ en la hoja del estudiante 2. Para esto, la función debe retornar $B = [[10], [0, 40], [1, 50], []]$.

Luego de irse el profesor 1, las hojas de los estudiantes se vuelven a intercambiar de acuerdo a P . Después del intercambio, las hojas son $A = [[10], [1, 50], [], [0, 40]]$

Finalmente el evaluador llama a: Finally, the grader calls:

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

Esta función debe retornar el arreglo $D = [1, 0, -1, 1]$ ya que los estudiantes 0 y 3 levantaron la mano en la ronda 1, el estudiante 1 levantó la mano en la ronda 0, y el estudiante 2 nunca levantó

la mano. En este ejemplo, $C = 3$.

Evaluador de ejemplo

Formato de entrada:

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

Aquí, Q es un arreglo de longitud N , donde $Q[i] = j$ si el estudiante i levanta la mano durante la ronda j , o $Q[i] = -1$ si el estudiante i nunca levanta la mano durante el juego completo.

Formato de salida:

Después de cada llamada a `process_step`, el evaluador de ejemplo imprime el contenido de las hojas. Sea K la longitud de B y $L[i]$ la longitud de $B[i]$ (para cada $0 \leq i < K$). El evaluador de ejemplo imprime B en el formato siguiente, **seguido de una línea en blanco**:

```
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]
```

Luego, el evaluador de ejemplo intercambia las hojas de acuerdo a la permutación P . Si $K \neq N$, el evaluador de ejemplo imprime un mensaje de error y se termina la ejecución.

Finalmente, después de llamar a `determine_steps`, el evaluador de ejemplo imprime:

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

Siendo H la longitud del arreglo D retornado por `determine_steps`.