

Juego en el Aula

Un prestigioso colegio de Tashkent celebra su aniversario organizando un juego entre alumnos y profesores. Hay N estudiantes sentados en el aula, numerados del 0 al $N - 1$. Cada estudiante sostiene una hoja de papel que contiene un arreglo de números enteros. Inicialmente, cada hoja está en blanco, por lo que contiene un arreglo vacío.

Los estudiantes están jugando con M profesores, numerados del 0 al $M - 1$, y el director. El juego se juega en M pasos, también enumerados del 0 al $M - 1$. En el paso j , el profesor j entra al aula y tienen lugar los siguientes eventos:

- Algunos estudiantes (posiblemente ninguno) levantan la mano. Se garantiza que en cada paso al menos un estudiante **no** levanta la mano, y cada estudiante puede levantar la mano **como máximo una vez** durante todo el juego.
- El profesor revisa los papeles que los alumnos tienen en sus manos y puede **modificar** los papeles de los alumnos que **no** levantaron la mano en este paso. Para cada alumno, el maestro puede reemplazar el contenido de su papel con un nuevo arreglo (posiblemente vacío) de números enteros. Cada número debe estar entre 0 y 63, inclusive, y el arreglo puede tener como máximo 63 elementos.
- Después de estas modificaciones, el profesor j se va y los estudiantes intercambian sus trabajos según una permutación secreta P . Es decir, cada estudiante i ($0 \leq i < N$) le da su papel al estudiante $P[i]$, donde $P[0], P[1], \dots, P[N - 1]$ son N enteros **distintos** entre 0 y $N - 1$, ambos inclusive. Los valores de P están **fijos** durante todas las etapas del juego, pero los profesores y el director no los conocen.

Una vez que todos los M pasos hayan terminado y se haya realizado el último intercambio según P , entra el director. Mirando solo los papeles que sostienen los estudiantes, el director debe determinar, para cada estudiante i ($0 \leq i < N$), el número de paso j en el que el estudiante i levantó la mano, o concluir que el estudiante i nunca levantó la mano.

Los profesores no pueden comunicarse entre sí ni con el director, salvo mediante los arreglos escritos en los papeles. Cada profesor sabe en qué paso entra al aula.

Tu tarea consiste en diseñar e implementar una estrategia para los profesores y el director que permita identificar correctamente si cada alumno levantó la mano y cuándo lo hizo. Tu puntuación dependerá del tamaño máximo de cualquier arreglo escrito en un papel: un tamaño máximo menor dará como resultado una mayor o igual puntuación.

Detalles de Implementación

Necesitas implementar dos funciones, una para los profesores y otro 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)
```

(**process_step** significa "procesar paso")

- N : número de estudiantes.
- M : número de pasos, y también el número de profesores.
- R : el número de paso actual (de 0 a $M - 1$).
- T : un arreglo (posiblemente vacío) que consta de los índices de los estudiantes que levantan la mano en este paso, en orden ascendente.
- A : un arreglo de tamaño N que describe los papeles, donde $A[i]$ ($0 \leq i < N$) es el arreglo de enteros en el papel que sostiene el estudiante i al comienzo del paso.
- Esta función se llama exactamente M veces por juego, en el orden $R = 0, 1, \dots, M - 1$.

La función debe retornar un arreglo B de tamaño N , representando los papeles después de las modificaciones del profesor, en el mismo formato que A .

- Por cada estudiante i que levantó la mano (es decir, i aparece en T), el papel no debe cambiarse, por lo que $B[i] = A[i]$.
- Para cada i tal que $0 \leq i < N$, el tamaño de $B[i]$ no debe exceder 63, y cada entero en $B[i]$ debe estar entre 0 y 63, ambos 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)
```

(**determine_steps** significa "determinar pasos")

- N, M : lo mismo antes definido.
- A : un arreglo de tamaño N , donde $A[i]$ es el arreglo de enteros en el papel que sostiene el estudiante i después de todos los M pasos.
- Esta función se llama exactamente una vez por juego, después de la llamada final a `process_step`.

La función debe retornar un arreglo D de tamaño N . Para cada i tal que $0 \leq i < N$, debe cumplirse que

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

Su programa no puede almacenar ni transmitir información de una llamada a `process_step` a otra, excepto a través del valor de retorno de `process_step`. Si su programa intenta hacerlo, su puntuación en CMS podría ser incorrecta y podría reducirse al finalizar el concurso. Tenga en cuenta que el evaluador puede ejecutar varias instancias de su programa simultáneamente. Esto significa que las llamadas a `process_step` y `determine_steps` del mismo juego pueden ejecutarse en instancias diferentes, y las llamadas a `process_step` y `determine_steps` de diferentes juegos pueden ejecutarse en la misma instancia. Cada caso de prueba incluye hasta 5 juegos.

Restricciones

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Cada estudiante levanta la mano **máximo una sola vez** durante todo el juego. En otras palabras, para cada estudiante i , hay como máximo un paso j en el que i levante la mano.
- En cada paso, al menos un estudiante **no** levanta la mano.

Subtasks and Scoring

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

Para cada caso de prueba, su puntuación es 0 (notificado como `Output isn't correct` en CMS) si el valor de retorno de cualquier llamada a la función `process_step` no es válido o si el valor de retorno de cualquier llamada a la función `determine_steps` es incorrecta.

De lo contrario, sea C el tamaño máximo de **cualquier** arreglo en **cualquier** B retornada por `process_step`. Entonces, la puntuación para un caso de prueba en una subtearea con puntuación S se calcula como $S \cdot X$, donde X depende de C según el sistema de tabla.

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

Ejemplos

Considere un escenario con $N = 4$ estudiantes, $M = 2$ profesores, y una permutación $P = [0, 3, 1, 2]$. Inicialmente, todo los papeles estan vacíos.

El evaluar primero llamada:

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

El estudiante 1 ha levantado su mano, entonces su papel no puede ser modificado. El profesor 0 puede decidir escribir $[10]$ en el papel del estudiante 0, escribiendo $[1, 63, 4]$ en el papel del estudiante 3, y dejar el papel del estudiante 2's vacío. Para hacer esto, la función retorna $B = [[10], [], [], [1, 63, 4]]$.

Después de que el profesor 0 se va, los estudiantes intercambian sus papeles según P . Después del intercambio, los papeles son $A = [[10], [], [1, 63, 4], []]$.

El evaluador llama a :

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

Los estudiantes 0 y 3 han levantado la mano, por lo que sus papeles no pueden ser modificados. El profesor 1 puede decidir escribir $[0, 40]$ en el papel del estudiante 1 y escribir $[1, 50]$ en el papel del estudiante 2. Para hacer esto, la función debe retornar $B = [[10], [0, 40], [1, 50], []]$.

Después de que el profesor 1 se va, los papeles se intercambian según P . Después del intercambio, los papeles son $A = [[10], [1, 50], [], [0, 40]]$.

Finalmente, el evaluador llama:

```
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 el paso 1, el estudiante 1 alzó 0, y el estudiante 2 nunca levantó su mano. En este ejemplo $C = 3$.

Sample Grader (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 tamaño N , donde $Q[i] = j$ si el estudiante i levanta la mano durante el paso j , o $Q[i] = -1$ si el estudiante i nunca levanta la mano durante todo el juego.

Formato de salida:

Tras cada llamada a `process_step`, el evaluador de ejemplo muestra el contenido de los papeles. Sea K el tamaño de B y $L[i]$ el tamaño de $B[i]$ (para cada $0 \leq i < K$). El calificador de muestra imprime B en el siguiente formato, **seguido por una línea vacía**:

```
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]
```

El evaluador de ejemplo luego intercambia los papeles según la permutación P . Si $K \neq N$, el evaluador de ejemplo imprime un mensaje de error y finaliza.

Después de llamar a `determine_steps`, el evaluador de ejemplo devuelve:

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

Aquí, H es el tamaño del arreglo D retornado por `determine_steps`.