

課堂遊戲

塔什干一所著名高中舉辦師生對抗賽，慶祝建校週年。教室裡坐著 N 位學生，編號從 0 到 $N - 1$ 。每個學生手裡拿著一張紙，紙上寫著一個整數陣列。最初，每張紙都是空白的，所以它包含的是一個空數組。

學生們正在和 M 位老師，編號從 0 到 $M - 1$ 以及一位校長玩遊戲。遊戲以 M 步驟進行，步數也從 0 到 $M - 1$ 編號。在步驟 j 中，教師 j 進入教室，並發生以下事件：

- 可能有部分（甚至沒有）學生舉手。可以保證，每一步至少有一名學生不舉手，並且每位學生在整個遊戲過程中最多只能舉手一次。
- 教師查看學生手中的紙張，並可修改未舉手學生的紙張的內容。對於每位未舉手的學生，教師可以將他們紙張上的內容替換為一個新的（可能為空的）整數數組。每個整數必須介於 0 和 63 之間（含 0 和 63 ），且陣列最多只能包含 63 個元素。
- 經過這些修改後，老師 j 離開，學生們依照一個秘密排列 P 交換紙張。也就是說，每個學生 i ($0 \leq i < N$) 將自己的紙張交給學生 $P[i]$ ，其中 $P[0], P[1], \dots, P[N - 1]$ 是介於 0 和 $N - 1$ 之間的 N 個不同的整數（包含 0 及 $N - 1$ ）。 P 的值在整個遊戲過程中都是固定的，但老師和校長都不知道這些值。

當所有 M 個步驟完成且根據 P 進行的最後一次交換也完成後，校長進入。校長只能查看學生手中的紙張，必須確定每位學生 i ($0 \leq i < N$) 舉手的步驟編號為 j ，或得出結論：學生 i 從未舉過手。

教師們無法與其他教師或校長交流，只能透過學生互相傳遞的紙條上所寫的整數陣列進行溝通。每個教師都知道他自己是第幾個老師進入課室。

你的任務是為教師和校長設計並實施一項策略，以準確識別每位學生舉手的時間和方式。你的得分將取決於紙上所寫數組的最大長度數組，其長度越短所得分越高或相同。

實作細節

你需要實施兩項程序，一項針對教師，一項針對校長。

你應該實現給**教師**的程式是：

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

- N ：學生人數。

- M : 步數，也是教師人數。
- R : 目前步驟編號 (從 0 到 $M - 1$) 。
- T : 一個 (可能為空的) 數組，包含在此步驟中舉手的學生的索引，以升序排列
- A : 長度為 N 的數組，用於描述紙張，其中 $A[i]$ ($0 \leq i < N$) 是學生 i 在步驟開始時所持有的紙張上的整數數組。
- 此過程每局遊戲調用 M 次，順序為 $R = 0, 1, \dots, M - 1$ 。

過程應傳回長度為 N 的陣列 B ，表示教師修改後的紙張，格式與 A 相同。

- 每位舉手的學生 i (即 i 出現在 T)，紙張不得更改，因此 $B[i] = A[i]$ 。
- 對於每個 i 使得 $0 \leq i < N$ ， $B[i]$ 的長度不得超過 63，且 $B[i]$ 中的每個整數都必須在 0 和 63。

你應該實現給校長的程式是：

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

- N, M : 與上述相同。
- A : 長度為 N 的數組，其中 $A[i]$ 是學生 i 在所有 M 步驟之後所持紙張上的整數數組。
- 此過程在每個遊戲中只調用一次，在最後一次調用 `process_step` 之後調用。

過程必須傳回長度為 N 數組 D 對於每個滿足 $0 \leq i < N$ 的 i ，必須滿足以下條件：

- 若學生 i 在步驟 j 期間舉手，則 $D[i] = j$ ，或者
- 如果學生 i 在整個遊戲過程中從未舉手， $D[i] = -1$ 。

您的程式不允許儲存或傳遞任何訊息，除了透過 `process_step` 的回傳值進行傳遞外。如果您的程式嘗試這樣做，您在 CMS 中的得分可能不正確，並且可能會在比賽結束後被降低。請注意，評分員可能會同時執行您程式的多個實例。這意味著，同一局遊戲中對 `process_step` 和 `determine_steps` 呼叫可能會在不同的實例中執行，而不同局遊戲中對 `process_step` 和 `determine_steps` 呼叫可能會在同一個實例中執行。每個測試案例最多包含 5 遊戲。

約束條件

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- 在整個遊戲過程中，每位學生最多舉手一次。換句話說，對於每位學生 i ，最多只有一步 j 中 i 會舉手。
- 在每個步驟中，至少有一名學生不舉手。

子任務和評分

子任務	分數	附加限制條件
1	4	$M = 1$
2	6	$N = 2$
3	9	對所有 $0 \leq i \leq N - 1, P[i] = i$ 。
4	25	每一步最多只有一名學生舉手。
5	56	無其他限制。

對於每個測試案例，如果對過程 `process_step` 的任何呼叫返回值無效，或者對過程 `determine_steps` 的任何呼叫返回值不正確，則您的得分為 0（在 CMS 中報告為 `Output isn't correct`）。

否則，令 C 為 `process_step` 傳回的任 B 中任意數組的最大長度。然後，子任務中得分為 S 測試案例的得分計算為 $S \cdot X$ ，其中 X 根據下表取決於 C ：

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

例子

考慮一個場景，其中 $N = 4$ 名學生， $M = 2$ 教師，排列 $P = [0, 3, 1, 2]$ 。最初，所有紙張都是空白的。

評分員首先呼叫：

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

學生 1 舉手了，所以他的紙張不能修改。教師 0 可以決定在學生 0 的紙張上寫 [10]，在學生 3 的紙張上寫 [1, 63, 4]，並將學生 2 的紙張留空。為此，該過程必須傳回 $B = [[10], [], [], [1, 63, 4]]$ 。

老師 0 離開後，學生們依照 P 交換紙張。交換後，紙張為 $A = [[10], [], [1, 63, 4], []]$ 。

評分員現在呼叫：

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

學生 0 和 3 舉手示意，因此他們的紙張不能修改。教師 1 可以決定在學生 1 的紙張上寫 $[0, 40]$ ，在學生 2 的紙張上寫 $[1, 50]$ 。為此，該過程必須傳回 $B = [[10], [0, 40], [1, 50], []]$ 。

老師 1 離開後，紙張再次依照 P 交換。交換後，紙張為 $A = [[10], [1, 50], [], [0, 40]]$ 。

最後，評分員呼叫：

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

此過程必須返回數組 $D = [1, 0, -1, 1]$ 因為學生 0 和 3 在步驟 1 中舉手，學生 1 在步驟 0 中舉手，而學生 2 從未舉手。在本例中， $C = 3$ 。

樣例評分程式

輸入格式：

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

這裡， Q 是一個長度為 N 的數組，其中 $Q[i] = j$ 如果學生 i 在步驟 j 舉手，或者 $Q[i] = -1$ 如果學生 i 在整個遊戲中從未舉手。

輸出格式：

每次呼叫 `process_step` 後，範例評分器都會輸出紙張的內容。設 K 為 B 的長度， $L[i]$ 為 $B[i]$ 的長度（對於每個 $0 \leq i < K$ ）。範例評分器以以下格式列印 B ，後面跟著一個空白行：

```
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]
```

然後，樣本評分員依照排列 P 交換紙張。如果 $K \neq N$ ，則樣本評分器將列印錯誤訊息並終止。

呼叫 `determine_steps` 後，範例評分器輸出：

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

這裡， H 是 `determine_steps` 傳回的陣列 D 的長度。