

課堂遊戲 (Classroom Game)

塔什干 (Tashkent) 一所知名高中為慶祝校慶，舉辦了一場師生對抗賽。教室裡坐著 N 名學生，編號從 0 到 $N - 1$ 。每位學生手持一張紙條，上面寫著一個整數陣列。遊戲開始時，每張紙條都是空白的，代表陣列為空。

學生們與 M 名教師（編號從 0 到 $M - 1$ ）以及校長進行這場遊戲。遊戲分成 M 個階段進行，這些階段同樣從 0 到 $M - 1$ 編號。在階段 j 中，老師 j 走進教室，並發生以下事件：

- 部分（可能為零個）學生舉手。可以保證在每個階段中，至少有一名學生**不會**舉手，且每位學生在整個遊戲過程中**至多**舉手一次。
- 老師檢視學生目前持有的紙條，並可**修改**本階段中**未**舉手的學生所持有的紙條內容。對於每位符合條件的學生，老師可將其紙條內容替換為一個新的（可能為空）整數陣列。每個整數必須介於 0 至 63 之間（含邊界），且該陣列最多可包含 63 個元素。
- 完成這些修改後，老師 j 離開，然後學生們則依照秘密排列 P 交換紙條。也就是說，每位學生 i ($0 \leq i < N$) 將自己的紙條交給學生 $P[i]$ ，其中 $P[0], P[1], \dots, P[N - 1]$ 是介於 0 至 $N - 1$ 之間（含邊界）的 N 個**兩兩相異**的整數。陣列 P 的值在遊戲的所有階段中皆**固定**，但老師們與校長並不知曉這些數值。

當所有的 M 個階段完成，且根據排列 P 進行了最後一次交換後，校長便會進入教室。校長僅憑觀察學生手中的紙條，必須針對每位學生 i ($0 \leq i < N$)，判定學生 i 在哪一階段 j 舉手，或是判定該學生 i 從未舉手。

除了透過寫在紙條上的整數陣列之外，老師們無法與其他老師或校長溝通。此外，每位老師都知道自己進入教室當下的階段編號。

你的任務是幫助教師們與校長設計並實施一套策略，用以正確判斷每位學生是否曾舉手以及舉手的時間點。你的得分將取決於紙張上所寫的任何整數陣列的最大長度：最大長度越短，獲得的分數可能相同或更高。

實作細節

您需要實作兩個程序，一個是針對教師們的，另一個是針對校長的。

您應為**教師**實作下列程序：

```
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 ：階段總數，同時也是教師總數。
- R ：當前的階段編號（從 0 至 $M - 1$ ）。
- T ：一個（可能為空）的陣列，包含本階段中所有舉手的學生編號，遞增排序。
- A ：一個長度為 N 的陣列，用以描述所有的紙條。其中， $A[i]$ ($0 \leq i < N$) 表示該步驟開始時，學生 i 手持紙條上的整數陣列。
- 此程序在每一場遊戲中會被呼叫恰好 M 次，且順序為 $R = 0, 1, \dots, M - 1$ 。

該程序應回傳一個長度為 N 的陣列 B ，用以表示經過教師修改後的紙條們，其格式應與 A 相同。

- 對於每位舉手的編號為 i 的學生（即 i 出現在 T 中），紙條上的內容不得被修改，即 $B[i] = A[i]$ 。
- 對於每一位學生 i ($0 \leq i < N$)，陣列 $B[i]$ 的長度不得超過 63，且 $B[i]$ 中的每個整數必須介於 0 與 63（含邊界）。

您應為**校長**實作下列程序：

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

- N 、 M ：與前述相同。
- A ：一個長度為 N 的陣列，其中 $A[i]$ 代表歷經所有 M 個階段後，學生 i 手持紙條上的陣列內容。
- 此程序在每場遊戲中僅會被呼叫恰好一次，且是在最後一次呼叫 `process_step` 之後才被呼叫。

此程序必須回傳一個長度為 N 的陣列 D 。對於每一個 i ($0 \leq i < N$) 須滿足以下條件：

- $D[i] = j$ ，若學生 i 在階段 j 舉手；或者
- $D[i] = -1$ ，若學生 i 在整場遊戲中從未舉手。

您的程式不得在不同次呼叫 `process_step` 之間儲存或傳遞任何資訊，惟透過 `process_step` 的回傳值傳遞者除外。若您的程式試圖進行此類操作，您在 CMS 上的分數可能會不正確，並可能在比賽結束後被扣分。請注意，評分系統可能會同時執行您程式的多個執行個體。這意味著來自同一場遊戲的 `process_step` 和 `determine_steps` 呼叫，可能會在不同的執行個體中執行；而來自不同遊戲的 `process_step` 和 `determine_steps` 呼叫，則可能會在同一個執行個體中執行。每筆測試資料最多包含 5 場遊戲。

限制條件

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$

- 每位學生在整場遊戲過程中**至多舉手一次**。換言之，對於每位學生 i ，至多只有一個階段 j ，在該階段中學生 i 會舉手。
- 在每個階段中，至少有一名學生**不會舉手**。

子任務與評分

子任務	分數	額外限制條件
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ 對所有的 $0 \leq i \leq N - 1$ 。
4	25	每個步驟中至多只有一名學生舉手。
5	56	無額外限制。

對於每個測試案例，若任何一次呼叫程序 `process_step` 的回傳值無效，或任何一次呼叫程序 `determine_steps` 的回傳值不正確，您的得分即為 0（在 CMS 中顯示為 `Output isn't correct`）。

否則，令 C 為 `process_step` 所回傳的**所有** B 中**所有**陣列的最大長度。此時，分數為 S 的子任務中的測試資料之分數，計算方式為 $S \cdot X$ ，其中 X 取決於 C ，具體取值參照下列表格：

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

範例

考慮一個包含 $N = 4$ 名學生、 $M = 2$ 名教師，且排列 $P = [0, 3, 1, 2]$ 的情境。遊戲開始時，所有紙條皆為空白。

評分程式首先呼叫：

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

學生 1 在此階段舉手，因此不能修改他的紙條。教師 0 決定在學生 0 手持紙條上寫 [10]、在學生 3 手持的紙條上寫 [1, 63, 4]，並讓學生 2 的紙條保持空白。為此，該程序應回傳 $B = [[10], [], [], [1, 63, 4]]$ 。

在教師 0 離開後，學生們依照排列 P 交換紙條。在交換後，紙條上的內容表示為 $A = [[10], [], [1, 63, 4], []]$ 。

評分程式接著呼叫：

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

學生 0 與 3 在此階段舉手，所以他們手上的紙條不能被修改。教師 1 決定在學生 1 的紙條上寫 [0, 40]、在學生 2 的紙條上寫 [1, 50]。為此，程序應回傳 $B = [[10], [0, 40], [1, 50], []]$ 。

在教師 1 離開後，學生們再次依照排列 P 交換紙條。這次交換後，紙條上的內容表示為 $A = [[10], [1, 50], [], [0, 40]]$ 。

評分程式最後呼叫：

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

此程序應回傳陣列 $D = [1, 0, -1, 1]$ ，因為學生 0 與 3 在階段 1 舉手、學生 1 在階段 0 舉手且學生 2 從未舉手。在這個範例中， $C = 3$ 。

範例評分程式

輸入格式：

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

這裡 Q 為長度 N 的陣列，其中 $Q[i] = j$ 用以表示學生 i 在階段 j 舉手，而 $Q[i] = -1$ 表示學生 i 在整場遊戲中從未舉手。

輸出格式：

每次呼叫 `process_step` 之後，範例評分程式會輸出所有紙條的內容。令 K 為 B 的長度而 $L[i]$ 為 $B[i]$ 的長度（對於所有的 $0 \leq i < K$ ）。範例評分程式會以下列格式輸出 B ，後面接一個空行：

```
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]
```

接著，範例評分程式會根據排列 P 交換紙條順序。若 $K \neq N$ ，那麼範例評分程式將會輸出錯誤訊息並且終止執行。

在呼叫 `determine_steps` 之後，範例評分程式會輸出：

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

其中 H 是 `determine_steps` 所回傳的陣列 D 的長度。