

Élèves vs Enseignant

Une école réputée de Tachkent célèbre son anniversaire en organisant un match entre élèves et enseignants. Il y a N élèves assis dans la salle de classe, numérotés de 0 à $N - 1$. Chaque élève tient une feuille de papier contenant un tableau d'entiers. Initialement, chaque feuille est vierge, son tableau est donc vide.

Les élèves jouent au jeu avec M enseignants, numérotés de 0 à $M - 1$, et le directeur. Le jeu se joue en M étapes, également numérotées de 0 à $M - 1$. À l'étape j , l'enseignant j entre dans la salle de classe et les événements suivants se produisent :

- Certains élèves (potentiellement aucun) lèvent la main. Il est garanti qu'à chaque étape, au moins un élève ne lève **pas** la main et que chaque élève ne peut lever la main qu'**une seule fois** au maximum pendant toute la partie.
- L'enseignant examine les feuilles que les élèves tiennent actuellement et peut **modifier** celles des élèves qui **n'ont pas** levé la main à cette étape. Pour chaque élève concerné, l'enseignant remplace le contenu de sa feuille par un nouveau tableau d'entiers (éventuellement vide). Chaque entier doit être compris entre 0 et 63 inclus et le tableau peut contenir au maximum 63 éléments.
- Après ces modifications, l'enseignant j et les élèves échangent leurs copies selon une permutation secrète P . Autrement dit, chaque élève i ($0 \leq i < N$) donne sa copie à l'élève $P[i]$, où $P[0], P[1], \dots, P[N - 1]$ sont N entiers **distincts** compris entre 0 et $N - 1$ inclus. Les valeurs de P restent **inchangées** tout au long du jeu, mais ni les enseignants ni le directeur ne les connaissent.

Une fois que chacune des M étapes est terminée et que le dernier échange selon P a été effectué, le directeur entre. En se basant uniquement sur les feuilles tenues par les élèves, il doit déterminer, pour chaque élève i ($0 \leq i < N$), le numéro de l'étape j à laquelle l'élève i a levé la main, ou conclure que l'élève i n'a jamais levé la main.

Les enseignants ne peuvent communiquer avec les autres enseignants ni avec le directeur, sauf par le biais des tableaux d'entiers inscrits sur les feuilles. Chaque enseignant connaît l'étape à laquelle il entre dans la classe.

Votre tâche consiste à concevoir et à implémenter une stratégie permettant aux enseignants et au directeur d'identifier correctement si et quand chaque élève a levé la main. Votre score dépendra de la longueur maximale de tout tableau inscrit sur une feuille : une longueur maximale plus courte donne un meilleur score.

Détails d'implémentation

Vous devez implémenter deux fonctions, une pour les enseignants et une pour le directeur.

La fonction à implémenter pour les **enseignants** est la suivante :

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

- N : le nombre d'élèves.
- M : le nombre d'étapes et le nombre d'enseignants.
- R : le numéro de l'étape actuelle (de 0 à $M - 1$).
- T : un tableau (éventuellement vide) contenant les indices des élèves qui lèvent la main à cette étape, trié par ordre croissant.
- A : un tableau de longueur N décrivant les feuilles, où $A[i]$ ($0 \leq i < N$) est le tableau d'entiers sur la feuille détenue par l'élève i au début de l'étape.
- Cette fonction est appelée exactement M fois par partie, dans l'ordre $R = 0, 1, \dots, M - 1$.

La fonction doit renvoyer un tableau B de longueur N , représentant les feuilles après les modifications de l'enseignant, dans le même format que A .

- Pour chaque élève i qui a levé la main (c'est-à-dire que i apparaît dans T), la feuille ne peut pas être modifiée, donc $B[i] = A[i]$.
- Pour chaque i tel que $0 \leq i < N$, la longueur de $B[i]$ ne doit pas dépasser 63 et chaque entier dans $B[i]$ doit être compris entre 0 et 63 inclus.

La fonction à implémenter pour le **directeur** est la suivante :

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

- N, M : identique à ci-dessus.
- A : un tableau de longueur N , où $A[i]$ est le tableau d'entiers sur la feuille tenu par l'élève i après les M étapes.
- Cette fonction est appelée exactement une fois par partie, après le dernier appel à `process_step`.

La fonction doit renvoyer un tableau D de longueur N . Pour chaque i tel que $0 \leq i < N$, elle doit garantir que

- $D[i] = j$ si l'élève i a levé la main pendant l'étape j , ou
- $D[i] = -1$ si l'élève i n'a jamais levé la main pendant la partie.

Votre programme ne doit ni stocker ni transmettre d'informations entre deux appels à `process_step`, sauf via la valeur de retour de `process_step`. Si votre programme tente de le faire, votre score dans le CMS risque d'être incorrect et pourrait être réduit après la fin du concours. Veuillez noter que l'évaluateur (grader) peut exécuter plusieurs instances de votre programme simultanément. Cela signifie que les appels à `process_step` et `determine_steps` d'une même partie peuvent être exécutés dans des instances différentes, et que les appels à `process_step` et `determine_steps` de parties différentes peuvent être exécutés dans la même instance. Chaque fichier test comprend jusqu'à 5 parties.

Contraintes

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Chaque élève lève la main **une seule fois au maximum** pendant toute la partie. Autrement dit, pour chaque élève i , il y a au plus une étape j au cours de laquelle i lève la main.
- À chaque étape, au moins un élève **ne** lève **pas** la main.

Sous-tâches et score

Sous-tâche	Score	Contraintes supplémentaires
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ pour tout $0 \leq i \leq N - 1$.
4	25	À chaque étape, un seul élève au maximum lève la main.
5	56	Aucune contrainte supplémentaire.

Pour chaque test, votre score est 0 (signalé comme `Output isn't correct` dans le CMS) si la valeur de retour d'au moins un appel à la fonction `process_step` est invalide ou si la valeur de retour d'au moins un appel à la fonction `determine_steps` est incorrecte.

Sinon, soit C la longueur maximale parmi **tous** les tableaux dans **tous** les B renvoyés par `process_step`. Alors, le score d'un test dans une sous-tâche de score S est calculé comme $S \cdot X$, où X dépend de C selon le tableau suivant :

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

Exemple

Considérons un scénario avec $N = 4$ élèves, $M = 2$ enseignants et une permutation $P = [0, 3, 1, 2]$. Au départ, tous les papiers sont vides.

L'évaluateur (grader) appelle en premier :

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

L'élève 1 a levé la main, sa copie ne peut donc pas être modifiée. L'enseignant 0 peut décider d'écrire $[10]$ sur la feuille de l'élève 0, d'écrire $[1, 63, 4]$ sur la feuille de l'élève 3 et de laisser la feuille de l'élève 2 vide. Pour ce faire, la fonction doit renvoyer $B = [[10], [], [], [1, 63, 4]]$.

Après le départ du professeur 0, les élèves échangent leurs papiers selon P . Après l'échange, les papiers sont $A = [[10], [], [1, 63, 4], []]$.

L'évaluateur (grader) appelle maintenant :

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

Les élèves 0 et 3 ont levé la main, leurs copies ne peuvent donc pas être modifiées. L'enseignant 1 peut décider d'écrire $[0, 40]$ sur la feuille de l'élève 1 et $[1, 50]$ sur la feuille de l'élève 2. Pour ce faire, la fonction doit renvoyer $B = [[10], [0, 40], [1, 50], []]$.

Après le départ du professeur 1, les papiers sont à nouveau échangés selon P . Après l'échange, les papiers sont $A = [[10], [1, 50], [], [0, 40]]$.

Finalement, l'évaluateur (grader) appelle :

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

Cette fonction doit renvoyer le tableau $D = [1, 0, -1, 1]$ puisque les élèves 0 et 3 ont levé la main à l'étape 1, l'élève 1 a levé la main à l'étape 0 et l'élève 2 n'a jamais levé la main. Dans cet exemple,

$C = 3$.

Évaluateur d'exemple (sample grader)

Format d'entrée :

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

Ici, Q est un tableau de longueur N , où $Q[i] = j$ si l'élève i lève la main pendant l'étape j , ou $Q[i] = -1$ si l'élève i ne lève jamais la main pendant toute la partie.

Format de sortie :

À la suite de chaque appel à `process_step`, l'évaluateur (grader) affiche le contenu des feuilles. Soit K la longueur de B et $L[i]$ la longueur de $B[i]$ (pour chaque $0 \leq i < K$). L'évaluateur (grader) imprime B au format suivant, **suivi d'une ligne vide** :

```
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]
```

L'évaluateur (grader) échange ensuite les copies selon la permutation P . Si $K \neq N$, l'évaluateur (grader) affiche un message d'erreur et se termine.

Après l'appel à `determine_steps`, l'évaluateur (grader) affiche :

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

Ici, H est la longueur du tableau D renvoyé par `determine_steps`.