

Classroom Game

Ένα από τα πιο αναγνωρισμένα λύκεια στην Τασκένδη γιορτάζει τα γενέθλιά του κάνοντας έναν αγώνα μεταξύ μαθητών και καθηγητών. Υπάρχουν N μαθητές στην τάξη, αριθμημένοι από 0 έως και $N - 1$. Κάθε μαθητής κρατάει ένα κομμάτι χαρτί που περιέχει έναν πίνακα ακεραίων. Αρχικά, κάθε χαρτί είναι κενό, περιέχει δηλαδή έναν κενό πίνακα.

Εκτός από τους μαθητές, στο παιχνίδι παίζουν M καθηγητές, αριθμημένοι από 0 έως και $M - 1$, και ο διευθυντής. Το παιχνίδι αποτελείται από M βήματα, επίσης αριθμημένα από 0 έως και $M - 1$. Στο βήμα j , ο καθηγητής j μπαίνει στην τάξη και τότε συμβαίνουν τα εξής:

- Μερικοί μαθητές (πιθανώς και κανένας) σηκώνουν το χέρι τους. Είναι εγγυημένο ότι σε κάθε βήμα τουλάχιστον ένας μαθητής **δεν** σηκώνει το χέρι του και κάθε μαθητής μπορεί να σηκώσει το χέρι του **το πολύ μία φορά** κατά τη διάρκεια ολόκληρου του παιχνιδιού.
- Ο καθηγητής κοιτάζει τα χαρτιά που κρατούν οι μαθητές και μπορεί να **αλλάξει** τα χαρτιά των μαθητών που **δεν** σήκωσαν το χέρι τους σε αυτό το βήμα. Για κάθε τέτοιο μαθητή, ο καθηγητής μπορεί να αντικαταστήσει το περιεχόμενο του χαρτιού του με έναν νέο (πιθανώς κενό) πίνακα ακεραίων. Κάθε ακέραιος πρέπει να είναι μεταξύ 0 και 63, συμπεριλαμβανομένων, και ο πίνακας μπορεί να έχει το πολύ 63 στοιχεία.
- Μετά από αυτές τις αλλαγές, ο καθηγητής j φεύγει και οι μαθητές ανταλλάσσουν τα χαρτιά τους σύμφωνα με μια μυστική μετάθεση P . Δηλαδή, κάθε μαθητής i (όπου $0 \leq i < N$) δίνει το γραπτό του στον μαθητή $P[i]$, όπου $P[0], P[1], \dots, P[N - 1]$ είναι N **διακριτοί** ακέραιοι αριθμοί μεταξύ 0 και $N - 1$, συμπεριλαμβανομένων. Οι τιμές του P είναι **σταθερές** σε όλα τα βήματα του παιχνιδιού, αλλά οι καθηγητές και ο διευθυντής δεν τις γνωρίζουν.

Μόλις ολοκληρωθούν όλα τα M βήματα και πραγματοποιηθεί η τελευταία ανταλλαγή σύμφωνα με τη μετάθεση P , μπαίνει στην τάξη ο διευθυντής. Κοιτάζοντας μόνο τα χαρτιά που κρατούν οι μαθητές, ο διευθυντής πρέπει να προσδιορίσει, για κάθε μαθητή i (όπου $0 \leq i < N$), τον αριθμό του βήματος j στο οποίο ο μαθητής i σήκωσε το χέρι του ή να συμπεράνει ότι ο μαθητής i δεν σήκωσε ποτέ το χέρι του.

Οι καθηγητές δεν μπορούν να επικοινωνήσουν με άλλους καθηγητές ή τον διευθυντή, παρά μόνο μέσω των πινάκων ακεραίων αριθμών που είναι γραμμένοι στα χαρτιά των μαθητών. Κάθε καθηγητής γνωρίζει τον αριθμό του βήματος κατά το οποίο μπαίνει στην αίθουσα.

Ζητείται να σχεδιάσετε και να υλοποιήσετε μια στρατηγική για τους καθηγητές και τον διευθυντή, η οποία να προσδιορίζει σωστά εάν και πότε κάθε μαθητής σήκωσε το χέρι του. Η βαθμολογία σας θα εξαρτηθεί από το μέγιστο μήκος οποιουδήποτε πίνακα γραμμένου σε ένα χαρτί μαθητή: ένα μικρότερο μέγιστο μήκος οδηγεί σε υψηλότερη ή ίση βαθμολογία.

Λεπτομέρειες υλοποίησης

Πρέπει να υλοποιήσετε δύο συναρτήσεις, μία για τους καθηγητές και μία για τον διευθυντή.

Η συνάρτηση που θα υλοποιήσετε για τους **καθηγητές** είναι:

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

- N : το πλήθος των μαθητών.
- M : το πλήθος των βημάτων, και επίσης το πλήθος των καθηγητών.
- R : ο αριθμός του τρέχοντος βήματος (από 0 έως και $M - 1$).
- T : ένας (πιθανώς κενός) πίνακας που αποτελείται από τους δείκτες των μαθητών που σηκώνουν το χέρι τους σε αυτό το βήμα, ταξινομημένους σε γνησίως αύξουσα σειρά.
- A : ένας πίνακας μήκους N που περιγράφει τα χαρτιά των μαθητών, όπου $A[i]$ (για $0 \leq i < N$) είναι ο πίνακας ακεραίων στο χαρτί που κρατάει ο μαθητής i στην αρχή του βήματος.
- Αυτή η συνάρτηση καλείται ακριβώς M φορές ανά παιχνίδι, με τη σειρά $R = 0, 1, \dots, M - 1$.

Η συνάρτηση θα πρέπει να επιστρέψει έναν πίνακα B μήκους N , που να αναπαριστά τα περιεχόμενα των χαρτιών μετά τις αλλαγές του καθηγητή, στην ίδια μορφή με τον πίνακα A .

- Για κάθε μαθητή i που σήκωσε το χέρι του (δηλαδή, αν το i εμφανίζεται στο T), το χαρτί δεν πρέπει να αλλάξει, επομένως $B[i] = A[i]$.
- Για κάθε i τέτοιο ώστε $0 \leq i < N$, το μήκος του $B[i]$ δεν πρέπει να υπερβαίνει το 63, και κάθε ακέραιος αριθμός στο $B[i]$ πρέπει να είναι μεταξύ 0 και 63, συμπεριλαμβανομένων.

Η συνάρτηση που θα υλοποιήσετε για τον **διευθυντή** είναι:

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

- N, M : ίδια όπως παραπάνω.
- A : ένας πίνακας μήκους N , όπου $A[i]$ είναι ο πίνακας ακεραίων στο χαρτί που κρατάει ο μαθητής i μετά από όλα τα M βήματα.
- Αυτή η συνάρτηση καλείται ακριβώς μία φορά ανά παιχνίδι, μετά την τελευταία κλήση της συνάρτησης `process_step`.

Η συνάρτηση πρέπει να επιστρέψει έναν πίνακα D μήκους N . Για κάθε i τέτοιο ώστε $0 \leq i < N$, πρέπει να ισχύει ότι

- $D[i] = j$ αν ο μαθητής i σήκωσε το χέρι του κατά τη διάρκεια του βήματος j , ή
- $D[i] = -1$ αν ο μαθητής i δεν σήκωσε ποτέ το χέρι του καθ' όλη τη διάρκεια του παιχνιδιού.

Δεν επιτρέπεται στο πρόγραμμά σας να αποθηκεύει ή να μεταδίδει πληροφορίες από μια κλήση της `process_step` σε μια άλλη, παρά μόνο μέσω της τιμής επιστροφής της `process_step`. Εάν το πρόγραμμά σας επιχειρήσει να το κάνει αυτό, η βαθμολογία σας στο CMS ενδέχεται να είναι λανθασμένη και μπορεί να μειωθεί μετά το τέλος του διαγωνισμού. Σημειώστε ότι ο βαθμολογητής μπορεί να εκτελέσει πολλαπλά στιγμιότυπα του προγράμματός σας ταυτόχρονα. Αυτό σημαίνει ότι οι κλήσεις της `process_step` και `determine_steps` του ίδιου παιχνιδιού μπορούν να εκτελεστούν σε διαφορετικά στιγμιότυπα και οι κλήσεις των `process_step` και `determine_steps` διαφορετικών παιχνιδιών μπορούν να εκτελεστούν στο ίδιο στιγμιότυπο. Κάθε περίπτωση ελέγχου περιλαμβάνει το πολύ 5 παιχνίδια.

Περιορισμοί

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Κάθε μαθητής σηκώνει το χέρι του **το πολύ μία φορά** κατά τη διάρκεια ολόκληρου του παιχνιδιού. Με άλλα λόγια, για κάθε μαθητή i , υπάρχει το πολύ ένα βήμα j στο οποίο ο i σηκώνει το χέρι του.
- Σε κάθε βήμα, τουλάχιστον ένας μαθητής **δεν** σηκώνει το χέρι του.

Υποπροβλήματα και βαθμολόγηση

Υποπρόβλημα	Βαθμολογία	Περαιτέρω περιορισμοί
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ για κάθε $0 \leq i \leq N - 1$.
4	25	Το πολύ ένας μαθητής σηκώνει το χέρι του σε κάθε βήμα.
5	56	Χωρίς περαιτέρω περιορισμούς

Για κάθε περίπτωση ελέγχου, η βαθμολογία σας είναι 0 (ένδειξη `Output isn't correct` στο CMS) εάν η τιμή επιστροφής οποιασδήποτε κλήσης της `process_step` δεν είναι έγκυρη ή εάν η τιμή επιστροφής οποιασδήποτε κλήσης της `determine_steps` είναι εσφαλμένη.

Διαφορετικά, έστω C το μέγιστο μήκος **οποιουδήποτε** πίνακα σε **οποιοδήποτε** B που επιστρέφεται από την `process_step`. Τότε, η βαθμολογία για ένα υποπρόβλημα με βαθμολογία S υπολογίζεται ως $S \cdot X$, όπου η τιμή του X εξαρτάται από το C σύμφωνα με τον ακόλουθο πίνακα:

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

Παράδειγμα

Θεωρήστε ένα σενάριο με $N = 4$ μαθητές, $M = 2$ καθηγητές και μετάθεση $P = [0, 3, 1, 2]$. Αρχικά, όλα τα χαρτιά είναι κενά.

Ο βαθμολογητής πρώτα καλεί:

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

Ο μαθητής 1 έχει σηκώσει το χέρι του, επομένως το χαρτί του δεν μπορεί να αλλάξει. Ο καθηγητής 0 μπορεί να αποφασίσει να γράψει [10] στο χαρτί του μαθητή 0, να γράψει [1, 63, 4] στο χαρτί του μαθητή 3 και να αφήσει το χαρτί του μαθητή 2 κενό. Για να γίνει αυτό, η συνάρτηση πρέπει να επιστρέψει $B = [[10], [], [], [1, 63, 4]]$.

Αφού φύγει ο καθηγητής 0, οι μαθητές ανταλλάσσουν τα χαρτιά τους σύμφωνα με τον P . Μετά την ανταλλαγή, τα χαρτιά είναι $A = [[10], [], [1, 63, 4], []]$.

Ο βαθμολογητής τώρα καλεί:

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

Οι μαθητές 0 και 3 έχουν σηκώσει το χέρι τους, επομένως τα γραπτά τους δεν μπορούν να αλλάξουν. Ο καθηγητής 1 μπορεί να αποφασίσει να γράψει [0, 40] στο χαρτί του μαθητή 1 και να γράψει [1, 50] στο χαρτί του μαθητή 2. Για να γίνει αυτό, η συνάρτηση πρέπει να επιστρέψει $B = [[10], [0, 40], [1, 50], []]$.

Αφού φύγει ο καθηγητής 1, τα χαρτιά ανταλλάσσονται ξανά σύμφωνα με τον P . Μετά την ανταλλαγή, τα χαρτιά είναι $A = [[10], [1, 50], [], [0, 40]]$.

Τέλος, ο βαθμολογητής καλεί:

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

Αυτή η συνάρτηση πρέπει να επιστρέψει τον πίνακα $D = [1, 0, -1, 1]$, αφού οι μαθητές 0 και 3 σήκωσαν το χέρι τους στο βήμα 1, ο μαθητής 1 σήκωσε το χέρι του στο βήμα 0 και ο μαθητής 2 δεν σήκωσε ποτέ το χέρι του. Σε αυτό το παράδειγμα, $C = 3$.

Ενδεικτικός βαθμολογητής

Μορφή εισόδου:

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

Εδώ, Q είναι ένας πίνακας μήκους N , όπου $Q[i] = j$ αν ο μαθητής i σηκώνει το χέρι του κατά τη διάρκεια του βήματος j , ή $Q[i] = -1$ αν ο μαθητής i δεν σηκώνει ποτέ το χέρι του κατά τη διάρκεια ολόκληρου του παιχνιδιού.

Μορφή εξόδου:

Μετά από κάθε κλήση της συνάρτησης `process_step`, ο ενδεικτικός βαθμολογητής εμφανίζει τα περιεχόμενα των χαρτιών. Έστω K το μήκος του B και $L[i]$ το μήκος του $B[i]$ (για κάθε $0 \leq i < K$). Ο ενδεικτικός βαθμολογητής εκτυπώνει το B στην ακόλουθη μορφή, **ακολουθούμενο από μια κενή γραμμή**:

```
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]
```

Ο ενδεικτικός βαθμολογητής ανταλλάσσει στη συνέχεια τα χαρτιά σύμφωνα με τη μετάθεση P . Αν $K \neq N$, ο ενδεικτικός βαθμολογητής εκτυπώνει ένα μήνυμα σφάλματος και τερματίζει, αντί για τα παραπάνω.

Μετά την κλήση της συνάρτησης `determine_steps`, ο ενδεικτικός βαθμολογητής εκτυπώνει:

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

Εδώ, H είναι το μήκος του πίνακα D που επιστρέφεται από τη συνάρτηση `determine_steps`.