

Classroom Game

Một trường trung học danh tiếng ở Tashkent tổ chức lễ kỷ niệm ngày thành lập bằng cách tổ chức một trò chơi giữa học sinh và giáo viên. Có N học sinh, được đánh số từ 0 đến $N - 1$, đang ngồi trong lớp học. Mỗi học sinh cầm một tờ giấy có chứa một mảng các số nguyên. Ban đầu, mỗi tờ giấy đều trống, do đó tờ giấy chứa một mảng rỗng.

Các học sinh đang tham gia trò chơi với M giáo viên, được đánh số từ 0 đến $M - 1$, và với hiệu trưởng. Trò chơi diễn ra trong M bước, cũng được đánh số từ 0 đến $M - 1$. Trong bước j , giáo viên j bước vào lớp học và diễn ra các sự kiện sau đây:

- Một số (có thể là không) học sinh giơ tay. Đảm bảo rằng trong mỗi bước, ít nhất có một học sinh **không** giơ tay, và mỗi học sinh chỉ được giơ tay **tối đa một lần** trong suốt trò chơi.
- Giáo viên xem các tờ giấy mà học sinh đang cầm, và có thể **thay đổi** nội dung trên các tờ giấy của những học sinh **không** giơ tay trong bước này. Đối với mỗi học sinh như vậy, giáo viên có thể thay thế nội dung trên tờ giấy của học sinh đó bằng một mảng số nguyên mới (có thể trống). Mỗi số nguyên phải nằm trong khoảng từ 0 đến 63 (tính cả hai đầu), và mảng mới này có thể có tối đa 63 phần tử.
- Sau khi thực hiện các điều chỉnh này, giáo viên j rời đi và các học sinh trao đổi các tờ giấy của mình theo một phép hoán vị bí mật P . Cụ thể, mỗi học sinh i ($0 \leq i < N$) sẽ trao tờ giấy của mình cho học sinh $P[i]$, trong đó $P[0], P[1], \dots, P[N - 1]$ là N số nguyên **phân biệt** nằm trong khoảng từ 0 đến $N - 1$ (tính cả hai đầu). Các giá trị của P là **cố định** trong suốt tất cả các bước của trò chơi, nhưng các giáo viên và hiệu trưởng không biết các giá trị này.

Khi tất cả M bước đã hoàn tất và lần trao đổi cuối cùng theo P được thực hiện, hiệu trưởng bước vào. Chỉ nhìn vào các tờ giấy mà các học sinh đang cầm, hiệu trưởng phải xác định, đối với mỗi học sinh i ($0 \leq i < N$), bước j mà học sinh i đã giơ tay, hoặc kết luận rằng học sinh i chưa bao giờ giơ tay.

Giáo viên không được phép trao đổi với các giáo viên khác hoặc với hiệu trưởng, ngoại trừ việc thông qua các dãy số nguyên được viết trên các tờ giấy. Mỗi giáo viên đều biết tại bước nào họ bước vào lớp học.

Nhiệm vụ của bạn là thiết kế và cài đặt một chiến lược cho các giáo viên và hiệu trưởng để xác định chính xác liệu mỗi học sinh có giơ tay hay không và vào thời điểm nào. Điểm số của bạn sẽ phụ thuộc vào độ dài tối đa của bất kỳ dãy số nào được viết trên một tờ giấy: độ dài tối đa này càng ngắn có thể mang lại điểm số của bạn càng cao hoặc bằng.

Chi tiết cài đặt

Bạn cần cài đặt hai hàm, một dành cho các giáo viên và một dành cho hiệu trưởng.

Hàm bạn cần cài đặt cho **các giáo viên** là:

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

- N : số lượng học sinh.
- M : số lượng bước, và cũng là số lượng giáo viên.
- R : số hiệu bước hiện tại (từ 0 đến $M - 1$).
- T : một mảng (có thể rỗng) chứa các chỉ số của các học sinh giơ tay trong bước này, sắp xếp theo thứ tự tăng dần.
- A : một mảng kích thước N mô tả các tờ giấy, trong đó $A[i]$ ($0 \leq i < N$) là mảng các số nguyên trên tờ giấy mà học sinh i đang cầm vào đầu bước này.
- Hàm này gọi đúng M lần cho mỗi trò chơi, theo thứ tự $R = 0, 1, \dots, M - 1$.

Hàm này cần trả về một mảng B kích thước N , chứa các tờ giấy sau khi giáo viên chỉnh sửa, theo cùng định dạng như A .

- Đối với mỗi học sinh i đã giơ tay (tức là i xuất hiện trong T), tờ giấy không được thay đổi, do đó $B[i] = A[i]$.
- Đối với mỗi số nguyên i sao cho $0 \leq i < N$, kích thước của $B[i]$ không được vượt quá 63, và mọi số nguyên trong $B[i]$ phải nằm trong khoảng từ 0 đến 63, tính cả hai đầu.

Hàm bạn cần cài đặt cho **hiệu trưởng** là:

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

- N, M : giống như trên.
- A : một mảng kích thước N , trong đó $A[i]$ là mảng các số nguyên trên tờ giấy của học sinh i sau toàn bộ M bước.
- Hàm này được gọi đúng một lần cho mỗi trò chơi, sau lời gọi cuối cùng đến hàm `process_step`.

Hàm này phải trả về một mảng D kích thước N . Với mỗi i sao cho $0 \leq i < N$, mảng phải thỏa mãn điều kiện sau:

- $D[i] = j$ nếu học sinh i đã giơ tay trong bước j , hoặc
- $D[i] = -1$ nếu học sinh i chưa bao giờ giơ tay trong suốt trò chơi.

Chương trình của bạn không được phép lưu trữ hoặc truyền bất kỳ thông tin nào từ một lời gọi hàm `process_step` sang lời gọi khác, trừ khi thông qua giá trị trả về của hàm `process_step`. Nếu chương trình của bạn cố gắng thực hiện điều này, điểm số của bạn trên CMS có thể không chính xác và có thể bị trừ sau khi cuộc thi kết thúc. Lưu ý rằng trình chấm có thể chạy nhiều tiến trình của chương trình của bạn cùng lúc. Điều này có nghĩa là các lời gọi đến `process_step` và `determine_steps` từ cùng một trò chơi có thể được chạy trong các tiến trình khác nhau, và các lời gọi đến `process_step` và `determine_steps` từ các trò chơi khác nhau có thể được chạy trong cùng một tiến trình. Mỗi trường hợp test bao gồm tối đa 5 trò chơi.

Các ràng buộc

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Mỗi học sinh chỉ được giơ tay **tối đa một lần** trong suốt trò chơi. Nói cách khác, đối với mỗi học sinh i , chỉ có tối đa một bước j mà tại đó i giơ tay.
- Tại mỗi bước, ít nhất có một học sinh **không** giơ tay.

Các subtask và Chấm điểm

Subtask	Điểm	Các ràng buộc thêm
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ với mọi $0 \leq i \leq N - 1$.
4	25	Có tối đa một học sinh giơ tay tại mỗi bước.
5	56	Không có ràng buộc nào thêm.

Đối với mỗi trường hợp test, điểm số của bạn sẽ là 0 (được hiển thị dưới dạng `Output isn't correct` trong CMS) nếu giá trị trả về của bất kỳ lời gọi nào đến hàm `process_step` là không hợp lệ hoặc nếu giá trị trả về của bất kỳ lời gọi nào đến hàm `determine_steps` là không chính xác.

Trái lại, gọi C là kích thước lớn nhất của **bất kỳ** mảng nào trong **bất kỳ** B nào được trả về bởi `process_step`. Khi đó, điểm số cho một trường hợp test trong một subtask có điểm số S được tính theo công thức $S \cdot X$, trong đó X phụ thuộc vào C theo bảng sau:

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

Ví dụ

Xét một kịch bản với $N = 4$ học sinh, $M = 2$ giảng viên và phép hoán vị $P = [0, 3, 1, 2]$. Ban đầu, tất cả các tờ giấy đều trống.

Trình chấm trước tiên gọi:

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

Học sinh 1 đã giơ tay, do đó tờ giấy của em này không thể được chỉnh sửa. Giáo viên 0 có thể quyết định ghi $[10]$ vào tờ giấy của học sinh 0, ghi $[1, 63, 4]$ vào tờ giấy của học sinh 3, và để trống tờ giấy của học sinh 2. Để thực hiện điều này, hàm phải trả về $B = [[10], [], [], [1, 63, 4]]$.

Sau khi giáo viên 0 ra khỏi phòng, các học sinh trao đổi tờ giấy của mình theo phép P . Sau khi trao đổi, các tờ giấy sẽ là $A = [[10], [], [1, 63, 4], []]$.

Trình chấm bây giờ gọi:

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

Các học sinh 0 và 3 đã giơ tay, do đó tờ giấy của các em này không thể được sửa đổi. Giáo viên 1 có thể quyết định ghi $[0, 40]$ vào tờ giấy của học sinh 1 và ghi $[1, 50]$ vào tờ giấy của học sinh 2. Để thực hiện điều này, hàm phải trả về kết quả $B = [[10], [0, 40], [1, 50], []]$.

Sau khi giáo viên 1 ra khỏi phòng, các tờ giấy lại được trao đổi theo quy tắc P . Sau khi trao đổi, các tờ giấy sẽ là $A = [[10], [1, 50], [], [0, 40]]$.

Cuối cùng, trình chấm gọi:

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

Hàm này phải trả về mảng $D = [1, 0, -1, 1]$ vì các học sinh 0 và 3 đã giơ tay ở bước 1, học sinh 1 đã giơ tay ở bước 0, và học sinh 2 chưa bao giờ giơ tay. Trong ví dụ này, $C = 3$.

Trình chấm mẫu

Định dạng dữ liệu vào:

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

Trong đó, Q là một mảng kích thước N , với $Q[i] = j$ nếu học sinh i giữ tay trong bước j , hoặc $Q[i] = -1$ nếu học sinh i không bao giờ giữ tay trong suốt trò chơi.

Định dạng kết quả ra:

Sau mỗi lời gọi hàm `process_step`, trình chấm mẫu sẽ in ra nội dung của các tờ giấy. Gọi K là kích thước của B và $L[i]$ là kích thước của $B[i]$ (với mỗi $0 \leq i < K$). Trình chấm mẫu sẽ in ra B theo định dạng sau (**kèm theo một dòng trống**):

```
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]
```

Trình chấm mẫu tiếp theo sẽ hoán vị các tờ giấy theo phép hoán vị P . Nếu $K \neq N$, trình chấm mẫu sẽ hiển thị thông báo lỗi và kết thúc ngay lập tức.

Sau khi gọi hàm `determine_steps`, trình chấm mẫu sẽ hiển thị kết quả:

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

Trong đó, H là kích thước của mảng D được trả về bởi `determine_steps`.