

Bài A. DOUBLEGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 giây

Cho một đồ thị có hướng không có chu trình. Đỉnh thứ i của đồ thị có chứa a_i đồng xu. Hai người chơi một trò chơi như sau:

- Hai người chơi luân phiên nhau thực hiện nước đi
- Đến lượt mình, người chơi chọn một đồng xu bất kỳ và di chuyển nó xuống một trong các đỉnh kề của đỉnh đang chứa đồng xu đó. Tất nhiên là không thể chọn đồng xu đang ở nút không thể di chuyển được nữa. Đồng xu sau khi được di chuyển sẽ tự động nhân đôi lên. Có nghĩa là nếu bạn chuyển một đồng xu từ x sang y , a_x sẽ giảm một còn a_y sẽ tăng hai
- Ai không thực hiện được nước đi hợp lệ nữa sẽ thua cuộc. Rõ ràng là trò chơi sẽ kết thúc sau hữu hạn bước, nên sẽ không có kết quả hòa

Biết rằng hai người chơi đều rất thông minh, hãy xác định xem liệu người đi trước có chiến thắng hay không

Dữ liệu vào

- Dòng đầu tiên chứa n, m : số đỉnh và số cung của đồ thị
- m dòng tiếp theo mỗi dòng chứa một cung: $u \ v$
- Dòng tiếp theo chứa $a_1, a_2, a_3, \dots, a_n$

Kết quả

Ghi YES/NO tương ứng là người đi trước thắng/thua

Ví dụ

stdin	stdout
5 4 1 2 2 3 3 4 4 5 5 4 3 3 1	YES

Hạn chế

- $1 \leq n, m, a_i \leq 10^5$

Bài B. ROOKGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Trò chơi ROOK được mô tả như sau:

- Cho bàn cờ n dòng m cột, trên đó có k quân xe (không có màu). Quân thứ i ở dòng x_i và cột y_i . Bàn cờ này hơi khác so với thực tế, mỗi ô có thể chứa được nhiều quân cờ cùng lúc, và luật di chuyển của xe trong trường hợp này là: Chỉ được phép di chuyển lên phía trên (cùng cột) hoặc sang bên trái (cùng dòng), ít nhất là một bước và nhiều nhất là đi đến cạnh của bàn cờ
- Có hai người chơi, luân phiên nhau thực hiện nước đi
- Đến lượt mình, người chơi chọn một quân xe tùy ý trên bàn và bỏ nó ra khỏi bàn cờ, hoặc là di chuyển nó sang một ô khác (theo luật di chuyển ở trên)
- Ai không thực hiện được nước đi hợp lệ nữa sẽ thua cuộc. Rõ ràng là trò chơi sẽ kết thúc sau hữu hạn bước, nên sẽ không có kết quả hòa

Biết rằng 2 người chơi đều rất thông minh, hãy xác định xem liệu người đi trước có thắng hay không

Dữ liệu vào

Gồm 5 testcase, mỗi testcase trên hai dòng:

- Dòng đầu chứa: n m k
- Dòng thứ hai chứa: x_1 y_1 x_2 y_2 \dots x_k y_k

Kết quả

Gồm 5 dòng, ghi YES/NO tương ứng là người đi trước thắng/thua

Ví dụ

stdin	stdout
8 8 3	NO
1 1 1 2 1 3	YES
8 8 3	YES
1 2 1 3 1 4	NO
100 100 1	NO
100 100	
10 10 2	
8 10 10 8	
10 10 2	
9 9 10 10	

Hạn chế

- $1 \leq n, m \leq 10^{18}$, $1 \leq k \leq 10^5$
- Subtask 0: $n = 1$
- Subtask 1: $n, m \leq 100$
- Subtask 2: $n, m \leq 1000$
- Subtask 3: Ràng buộc gốc

Bài C. XGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Cho một cây có gốc có n đỉnh, các đỉnh được đánh số từ 1 đến n với gốc là đỉnh 1. Xét một trò chơi trên cây này như sau:

- Có hai người, luân phiên nhau thực hiện lượt chơi
- Đến lượt mình, người chơi chọn một cạnh của cây và xóa đi. Khi đó, toàn bộ cây con ứng với cạnh đó sẽ bị xóa (tức là những đỉnh không liên thông với gốc đều sẽ bị xóa)
- Ai không thực hiện được lượt chơi hợp lệ nữa sẽ thua cuộc. Rõ ràng là trò chơi sẽ kết thúc sau hữu hạn bước, nên sẽ không có kết quả hòa

Bạn sẽ chơi trò này với máy, bạn được quyền chọn người đi trước và hãy dành chiến thắng trong trò chơi.

Tương tác:

- Đầu tiên bạn cần đọc vào số n ($0 \leq n \leq 5000$)
- Tiếp theo bạn cần đọc vào $n - 1$ số, số thứ i là đỉnh cha của đỉnh $i + 1$
- Sau đó bạn cần in ra 1 hoặc 0 tương ứng là bạn muốn đi trước hoặc đi sau
- Sau đó trò chơi sẽ bắt đầu. Khi đến lượt máy, máy sẽ in ra một số và bạn cần phải đọc vào số này và chuyển sang lượt chơi của bạn. Khi đến lượt bạn, bạn cần in ra một số và chuyển sang lượt chơi của máy
- Số được người chơi in ra là chỉ số của đỉnh xa gốc hơn trong số hai đỉnh của cạnh muốn xóa (tức là nếu bạn in ra x thì bạn sẽ xóa cạnh nối x với cha của x)
- Trò chơi sẽ kết thúc khi không có cạnh nào có thể đến được từ gốc nữa. Lúc này bạn cần kết thúc chương trình của mình (không đọc vào cũng không in ra gì nữa)

Lưu ý, sau mỗi lần in ra bạn cần đẩy dữ liệu ra luồng chuẩn (`flush(stdout)` hoặc `cout << endl`) để tương tác được với máy.

Ví dụ

<code>stdin</code>	<code>stdout</code>
8	1
1 1 2 2 3 3 4	8
4	5
3	2

Bài D. HGAME

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Có n viên sỏi sắp thẳng hàng, được đánh số từ 1 đến n từ trái sang phải. Ánh và Ngọc đang chơi một trò chơi như sau:

- Hai người họ luân phiên nhau thực hiện lượt chơi, Ngọc chơi trước
- Đến lượt mình, người chơi lấy đi một đoạn liên tiếp không quá 10 viên sỏi. Tức là họ chọn hai số tự nhiên L, H ($1 \leq L \leq n; L \leq H \leq \min(n, L + 9)$) sao cho tất cả các viên sỏi từ L đến H đều chưa bị lấy, sau đó họ lấy hết các viên sỏi từ L đến H
- Ai không thực hiện được lượt chơi hợp lệ nữa sẽ thua cuộc. Rõ ràng là trò chơi sẽ kết thúc sau hữu hạn bước, nên sẽ không có kết quả hòa

Hiện tại họ đã chơi được k lượt, bạn và máy sẽ tiếp tục trò chơi. Bạn được cho các thông tin về trò chơi và có quyền chọn người chơi lượt tiếp theo. Hãy dành chiến thắng

Tương tác:

- Đầu tiên bạn cần đọc vào hai số n, k
- Tiếp theo bạn cần đọc vào k cặp số, cặp thứ i là L_i, H_i : Vị trí bắt đầu và kết thúc của đoạn sỏi được lấy ở lượt chơi thứ i . Các cặp được liệt kê theo đúng thứ tự mà hai người chơi đã chơi
- Sau đó bạn cần in ra 1 hoặc 0 tương ứng là bạn sẽ chơi lượt tiếp theo hoặc máy sẽ chơi lượt tiếp theo
- Sau đó trò chơi sẽ duy trì. Khi đến lượt máy, máy sẽ in ra hai số, bạn cần phải đọc vào số này và chuyển sang lượt chơi của bạn. Khi đến lượt bạn, bạn cần in ra hai số và chuyển sang lượt chơi của máy
- Hai số được người chơi in ra là vị trí bắt đầu và kết thúc của đoạn sỏi sẽ lấy
- Trò chơi sẽ kết thúc khi cả n viên sỏi đều đã bị lấy. Lúc này bạn cần kết thúc chương trình của mình (không đọc vào cũng không in ra gì nữa)

Lưu ý, sau mỗi lần in ra bạn cần đẩy dữ liệu ra luồng chuẩn (flush(stdout) hoặc cout « endl) để tương tác được với máy.

Ví dụ

stdin	stdout
11 1	1
2 2	4 11
3 3	1 1

Hạn chế

- Nếu $k = 0$ thì $n \leq 10^5$
- Nếu $k > 0$ thì $n \leq 5000$