



Búp bê cơ khí

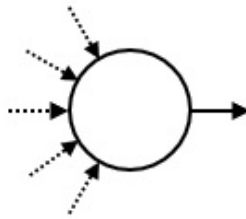
Búp bê cơ khí là một con búp bê tự động lặp đi lặp lại một dãy các cử động nhất định. Ở Nhật Bản, có nhiều loại búp bê cơ khí được tạo từ thời xưa.

Các cử động của búp bê cơ khí được điều khiển bởi một **mạch** bao gồm nhiều **thiết bị**. Các thiết bị được nối với nhau bởi các đường ống. Mỗi thiết bị có một hoặc hai **đầu ra**, và có thể có một số lượng tùy ý (có thể là không) **đầu vào**. Mỗi đường ống nối một đầu ra của một thiết bị với một đầu vào của chính thiết bị đó hoặc của một thiết bị khác. Chính xác một đường ống được nối với mỗi đầu vào, và chính xác một đường ống được nối với mỗi đầu ra.

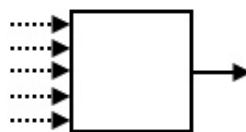
Để mô tả cách một con búp bê thực hiện các cử động, xét một **quả bóng** được đặt tại một trong các thiết bị. Quả bóng di chuyển trong mạch. Tại mỗi bước của quá trình di chuyển, quả bóng rời thiết bị qua một trong các đầu ra của thiết bị, di chuyển theo đường ống nối đến đầu ra và di chuyển đến thiết bị tại đầu kia của đường ống.

Có ba kiểu thiết bị: **nguồn (origin)**, **kích hoạt (trigger)**, và **chuyển đổi (switch)**. Có đúng một thiết bị **nguồn**, M thiết bị kích hoạt, và S thiết bị chuyển đổi (S có thể bằng không). Bạn phải xác định giá trị của S . Mỗi thiết bị có một số hiệu duy nhất.

Thiết bị nguồn là thiết bị nơi đầu tiên quả bóng được đặt vào. Thiết bị nguồn có một đầu ra. Số hiệu của nó là 0.

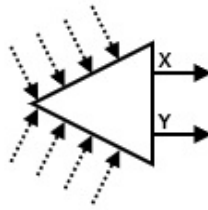


Một thiết bị kích hoạt sẽ điều khiển búp bê thực hiện một cử động cụ thể khi quả bóng di chuyển qua nó. Mỗi thiết bị kích hoạt có một đầu ra. Số hiệu của các thiết bị kích hoạt là từ 1 đến M .



Mỗi thiết bị chuyển đổi có 2 đầu ra được gọi là 'X' và 'Y'. **Trạng thái** của một thiết bị chuyển đổi hoặc là 'X' hoặc là 'Y'. Sau khi quả bóng di chuyển đến một thiết bị chuyển

đổi, nó rời thiết bị chuyển đổi sử dụng đầu ra được xác định bởi trạng thái hiện tại của thiết bị chuyển đổi. Sau đó, thiết bị chuyển đổi sẽ chuyển trạng thái của nó sang trạng thái đối lại. Ban đầu trạng thái của tất cả các thiết bị chuyển đổi là 'X'. Số hiệu của các thiết bị chuyển đổi là từ -1 đến $-S$.



Bạn được cho số lượng các thiết bị kích hoạt M . Bạn cũng được cho một dãy A có độ dài N , mỗi phần tử là số hiệu của một thiết bị kích hoạt. Mỗi thiết bị kích hoạt có thể xuất hiện một vài (có thể không) lần trong dãy A . Nhiệm vụ của bạn là tạo ra một mạch thỏa mãn các điều kiện sau đây:

- Quả bóng quay lại thiết bị nguồn sau một số bước nào đó.
- Khi quả bóng quay lại thiết bị nguồn lần đầu tiên, trạng thái của mỗi thiết bị chuyển đổi là 'X'.
- Quả bóng quay lại thiết bị nguồn lần đầu tiên sau khi di chuyển qua các thiết bị kích hoạt đúng N lần. Số hiệu của các thiết bị kích hoạt, theo thứ tự mà quả bóng di chuyển qua, là A_0, A_1, \dots, A_{N-1} .
- Gọi P là tổng số lượng thay đổi trạng thái của tất cả các thiết bị chuyển đổi do quả bóng gây ra trước khi nó di chuyển về thiết bị nguồn lần đầu tiên. Giá trị của P không được vượt quá 20 000 000.

Cùng lúc đó, bạn không muốn sử dụng quá nhiều thiết bị chuyển đổi.

Chi tiết cài đặt

Bạn phải cài đặt thủ tục sau đây.

```
create_circuit(int M, int[] A)
```

- M : số lượng thiết bị kích hoạt.
- A : mảng độ dài N , cho biết số hiệu của các thiết bị kích hoạt theo thứ tự mà quả bóng cần di chuyển qua.
- Thủ tục này được gọi đúng một lần.
- Lưu ý, giá trị N là độ dài của mảng A , và nó có thể thu được theo một trong các cách làm được mô tả trong Lưu ý cài đặt.

Chương trình của bạn phải gọi thủ tục sau đây để trả lời.

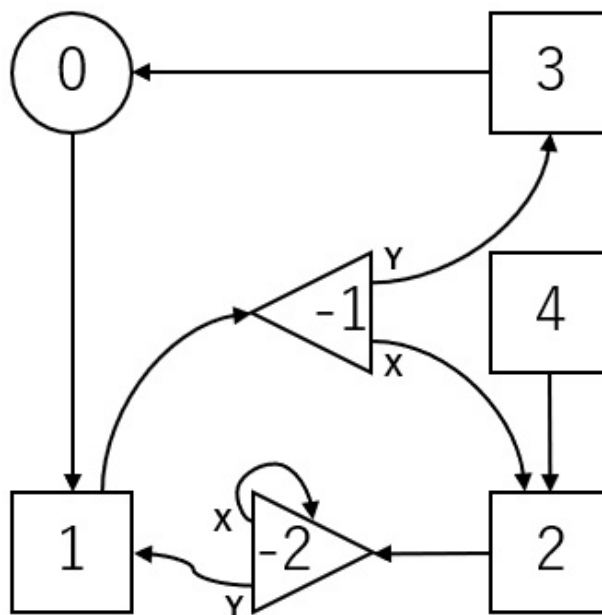
```
answer(int[] C, int[] X, int[] Y)
```

- C : mảng độ dài $M + 1$. Đầu ra của thiết bị i ($0 \leq i \leq M$) được nối với thiết bị $C[i]$.
- X, Y : Các mảng cùng độ dài. Độ dài S của các mảng này là số lượng các thiết bị chuyển đổi. Với thiết bị chuyển đổi $-j$ ($1 \leq j \leq S$), đầu ra 'X' của nó được nối với thiết bị $X[j - 1]$ và đầu ra 'Y' của nó được nối với thiết bị $Y[j - 1]$.
- Mọi phân tử của $C, X,$ và Y phải là các số nguyên nằm giữa $-S$ và M , bao gồm cả hai đầu mút.
- S nhiều nhất là 400 000.
- Thủ tục này phải được gọi đúng 1 lần.
- Mạch được biểu diễn bởi $C, X,$ và Y phải thỏa mãn các điều kiện trong phát biểu bài toán.

Nếu điều kiện nào đó trong các điều kiện nêu trên không được thỏa mãn, chương trình được chấm là **Wrong Answer**. Ngược lại, chương trình của bạn được chấm là **Accepted** và điểm của bạn được tính theo S (xem Subtasks).

Ví dụ

Cho $M = 4, N = 4$, và $A = [1, 2, 1, 3]$. Trình chấm gọi `create_circuit(4, [1, 2, 1, 3])`.



Hình trên biểu diễn một mạch, được mô tả bằng cách gọi hàm `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Các con số trong hình là số hiệu của các thiết bị.

Có hai thiết bị chuyển đổi được sử dụng. Do đó, $S = 2$.

Đầu tiên, trạng thái của cả hai thiết bị chuyển đổi -1 và -2 là 'X'.

Quả bóng di chuyển như sau:

$$0 \longrightarrow 1 \longrightarrow -1 \xrightarrow{X} 2 \longrightarrow -2 \xrightarrow{X} -2 \xrightarrow{Y} 1 \longrightarrow -1 \xrightarrow{Y} 3 \longrightarrow 0$$

- Khi quả bóng lần đầu tiên di chuyển đến thiết bị chuyển đổi -1 , trạng thái của nó là 'X'. Do đó, quả bóng di chuyển tới thiết bị kích hoạt 2. Tiếp theo, trạng thái của thiết bị chuyển đổi -1 được chuyển thành 'Y'.
- Khi quả bóng di chuyển đến thiết bị chuyển đổi -1 lần thứ hai, trạng thái của thiết bị là 'Y'. Do đó, quả bóng di chuyển đến thiết bị kích hoạt 3. Tiếp theo, trạng thái của thiết bị chuyển đổi -1 được chuyển thành 'X'.

Quả bóng lần đầu tiên quay lại thiết bị nguồn, đã di chuyển qua các thiết bị kích hoạt 1, 2, 1, 3. Trạng thái của cả hai thiết bị chuyển đổi -1 và -2 là 'X'. Giá trị của P là 4. Vì vậy, mạch thỏa mãn các điều kiện.

File `sample-01-in.txt` trong gói nén đính kèm tương ứng với ví dụ này. Dữ liệu đầu vào của các ví dụ khác cũng có trong gói đính kèm này.

Hạn chế

- $1 \leq M \leq 100\,000$
- $1 \leq N \leq 200\,000$
- $1 \leq A_k \leq M$ ($0 \leq k \leq N - 1$)

Subtasks

Điểm và các hạn chế cho mỗi test case như sau:

1. (2 điểm) Với mỗi i ($1 \leq i \leq M$), số nguyên i xuất hiện tối đa một lần trong dãy A_0, A_1, \dots, A_{N-1} .
2. (4 điểm) Với mỗi i ($1 \leq i \leq M$), số nguyên i xuất hiện tối đa hai lần trong dãy A_0, A_1, \dots, A_{N-1} .
3. (10 điểm) Với mỗi i ($1 \leq i \leq M$), số nguyên i xuất hiện tối đa 4 lần trong dãy A_0, A_1, \dots, A_{N-1} .
4. (10 điểm) $N = 16$
5. (18 điểm) $M = 1$
6. (56 điểm) Không có hạn chế gì thêm

Với mỗi test case, nếu chương trình của bạn được chấm là **Accepted**, điểm của bạn được tính dựa vào giá trị S .

- Nếu $S \leq N + \log_2 N$, bạn đạt toàn bộ số điểm của test case.
- Với mỗi test case trong Subtasks 5 và 6, nếu $N + \log_2 N < S \leq 2N$, bạn đạt một phần điểm. Điểm của test case này là $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$, nhân với số điểm được cho của subtask.
- Trái lại, điểm là 0.

Lưu ý, điểm của bạn cho mỗi subtask là điểm nhỏ nhất của các test case trong subtask.

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu đầu vào từ thiết bị vào chuẩn theo khuôn mẫu sau:

- Dòng 1: $M N$
- Dòng 2: $A_0 A_1 \dots A_{N-1}$

Trình chấm mẫu xuất ba dữ liệu ra.

Đầu tiên, trình chấm mẫu đưa câu trả lời của bạn ra file đặt tên là `out.txt` theo khuôn mẫu sau:

- Dòng 1: S
- Dòng $2 + i$ ($0 \leq i \leq M$): $C[i]$
- Dòng $2 + M + j$ ($1 \leq j \leq S$): $X[j - 1] Y[j - 1]$

Thứ hai, trình chấm mẫu mô phỏng các bước di chuyển của quả bóng. Nó đưa ra số hiệu của các thiết bị quả bóng di chuyển qua theo thứ tự ra file đặt tên `log.txt`.

Thứ ba, trình chấm mẫu in đánh giá kết quả của bạn ra thiết bị ra chuẩn .

- Nếu chương trình của bạn được chấm là **Accepted**, trình chấm mẫu in ra S và P theo khuôn mẫu `Accepted: S P`.
- Nếu chương trình của bạn được chấm là **Wrong Answer**, nó in ra `Wrong Answer: MSG`. Ý nghĩa của MSG như sau:
 - `answered not exactly once`: Thủ tục `answer` được gọi không phải một lần duy nhất.
 - `wrong array length`: Độ dài của C không là $M + 1$, hoặc độ dài của X và Y khác nhau.
 - `over 400000 switches`: S lớn hơn 400 000.
 - `wrong serial number`: Có một phần tử của C , X , hoặc Y nhỏ hơn $-S$ hoặc lớn hơn M .
 - `over 20000000 inversions`: Quả bóng không quay lại thiết bị nguồn trong phạm vi 20 000 000 lần chuyển trạng thái của các thiết bị chuyển đổi.
 - `state 'Y'`: Có một thiết bị chuyển đổi mà trạng thái là 'Y' khi quả bóng quay lại thiết bị nguồn lần đầu tiên.
 - `wrong motion`: Các thiết bị kích hoạt tạo ra các cử động khác với dãy A .

Lưu ý, trình chấm có thể không tạo ra `out.txt` và/hoặc `log.txt` khi chương trình của bạn được chấm là `Wrong Answer`.