



Prisoner Challenge

Trong một nhà tù có 500 tù nhân. Một ngày kia, quản ngục cho họ một cơ hội để tự giải thoát. Quản ngục để hai túi tiền, túi A và túi B, trong một căn phòng. Mỗi túi chứa một số lượng xu nằm trong khoảng từ 1 đến N , bao gồm cả hai đầu mút. Số lượng xu trong túi A **khác** số lượng xu trong túi B. Quản ngục đưa ra một thử thách cho các tù nhân. Mục tiêu của các tù nhân là xác định túi nào chứa ít xu hơn.

Trong căn phòng, cùng với hai túi tiền, còn có một bảng trắng. Lúc nào trên bảng cũng có ghi một số nguyên. Ban đầu, trên bảng ghi số 0.

Sau đó, quản ngục gọi lần lượt từng tù nhân vào trong phòng. Mỗi tù nhân khi vào phòng không biết ai hay bao nhiêu tù nhân khác đã vào phòng trước mình. Mỗi lần một tù nhân vào phòng, anh ta đọc số đang có trên bảng. Sau khi đọc xong, anh ta phải chọn một trong hai túi A hoặc B. Sau đó, tù nhân **kiểm tra** túi được chọn và biết được số lượng xu nằm trong túi đó. Tiếp theo, tù nhân phải thực hiện một trong hai **hành động** sau:

- Thay số được ghi trên bảng với một số nguyên không âm và rời khỏi căn phòng. Lưu ý rằng họ có thể thay đổi hoặc giữ nguyên số hiện tại. Sau đó, thử thách tiếp tục (trừ khi tất cả 500 tù nhân đều đã vào căn phòng).
- Xác định được túi nào là túi có ít xu hơn. Hành động này sẽ kết thúc thử thách.

Quản ngục sẽ không bao giờ yêu cầu một tù nhân đã rời khỏi phòng vào lại phòng lần thứ hai.

Các tù nhân sẽ giành chiến thắng thử thách này nếu một trong số họ xác định chính xác túi nào có ít xu hơn. Họ sẽ thua nếu bất kì ai trong số họ xác định sai, hoặc tất cả 500 tù nhân đều đã vào phòng và không xác định được túi có ít xu hơn.

Trước khi thử thách bắt đầu, các tù nhân tập trung trong sảnh chính của nhà tù và cùng quyết định một **chiến thuật** chung cho thử thách này với ba bước.

- Họ chọn một số nguyên không âm x , là số lớn nhất mà họ có thể viết lên bảng.
- Họ quyết định, với mỗi số i được viết lên bảng ($0 \leq i \leq x$), cái túi nào cần được kiểm tra nếu đọc được số i khi mới vào phòng.
- Họ quyết định, hành động nào tù nhân trong phòng sẽ thực hiện sau khi biết được số lượng xu trong túi được chọn. Cụ thể, với mỗi số i được viết trên bảng ($0 \leq i \leq x$) và với mỗi số j là số lượng xu trong túi được kiểm tra ($1 \leq j \leq N$), họ sẽ quyết định
 - một số nguyên trong khoảng 0 và x (bao gồm cả hai đầu mút) cần phải được viết lên bảng, hoặc

- cái túi nào là túi chứa ít xu hơn.

Nếu như giành chiến thắng thử thách này, quản ngục sẽ thả các tù nhân sau khi họ chịu án thêm x ngày nữa.

Bạn hãy tạo ra một chiến thuật cho các tù nhân để đảm bảo rằng họ sẽ giành chiến thắng thử thách này (bất kể số lượng xu trong túi A và túi B là bao nhiêu). Điểm của bạn sẽ phụ thuộc vào giá trị x (xem chi tiết ở phần Subtask).

Chi tiết cài đặt

Bạn cần cài đặt hàm sau:

```
int[][] devise_strategy(int N)
```

- N : là số lượng xu tối đa trong mỗi túi.
- Hàm này cần trả về một mảng s mà mỗi phần tử là một mảng chứa $N + 1$ số nguyên, thể hiện chiến thuật của bạn. Giá trị của x là độ dài của mảng s trừ một. Với mỗi số i mà $0 \leq i \leq x$, mảng $s[i]$ thể hiện tù nhân sẽ làm gì nếu họ đọc được số i ở trên bảng khi mới vào phòng:
 1. Giá trị $s[i][0]$ bằng 0 nếu tù nhân phải kiểm tra túi A, hoặc bằng 1 nếu tù nhân phải kiểm tra túi B.
 2. Gọi j là số lượng xu trong túi được kiểm tra. Tù nhân cần thực hiện hành động:
 - Nếu giá trị của $s[i][j]$ bằng -1 , tù nhân sẽ xác định túi A là túi có ít xu hơn.
 - Nếu giá trị của $s[i][j]$ bằng -2 , tù nhân sẽ xác định túi B là túi có ít xu hơn.
 - Nếu giá trị của $s[i][j]$ là một số không âm, tù nhân sẽ viết số đó lên bảng. Lưu ý rằng $s[i][j]$ không được vượt quá x .
- Hàm này sẽ được gọi đúng một lần.

Ví dụ

Xét lời gọi hàm sau:

```
devise_strategy(3)
```

Gọi v là số mà tù nhân đọc được trên bảng khi vào phòng. Một trong các chiến thuật đúng như sau:

- Nếu $v = 0$ (bao gồm cả số ban đầu), kiểm tra túi A.
 - Nếu túi A chứa 1 xu, xác định túi A là túi có ít xu hơn.
 - Nếu túi A chứa 3 xu, xác định túi B là túi có ít xu hơn.
 - Nếu túi A chứa 2 xu, viết số 1 lên bảng (xoá số 0).
- Nếu $v = 1$, kiểm tra túi B.
 - Nếu túi B chứa 1 xu, xác định túi B là túi có ít xu hơn.

- Nếu túi B chứa 3 xu, xác định túi A là túi có ít xu hơn.
- Nếu túi B chứa 2 xu, viết số 0 lên bảng (xoá số 1). Lưu ý rằng trường hợp này sẽ không bao giờ xảy ra vì chúng ta có thể khẳng định cả hai túi đều chứa 2 xu, đây là điều không được phép.

Chiến thuật này có thể được xác định bằng cách trả về $[[0, -1, 1, -2], [1, -2, 0, -1]]$. Độ dài của mảng được trả về là 2, vì vậy với giá trị trả về này thì giá trị của x là $2 - 1 = 1$.

Ràng buộc

- $2 \leq N \leq 5000$

Subtask

1. (5 điểm) $N \leq 500$, giá trị của x không được vượt quá 500.
2. (5 điểm) $N \leq 500$, giá trị của x không được vượt quá 70.
3. (90 điểm) Giá trị của x không được vượt quá 60.

Nếu trong bất cứ test case nào, mảng trả về bởi hàm `devise_strategy` không thể hiện một chiến thuật đúng, điểm cho lời giải của bạn với subtask đó sẽ là 0.

Trong subtask 3, bạn có thể có điểm lẻ. Gọi m là giá trị lớn nhất của x đối với tất cả các test case trong subtask này. Điểm của bạn cho subtask này sẽ được tính dựa trên bảng sau:

Điều kiện	Điểm
$40 \leq m \leq 60$	20
$26 \leq m \leq 39$	$25 + 1.5 \times (40 - m)$
$m = 25$	50
$m = 24$	55
$m = 23$	62
$m = 22$	70
$m = 21$	80
$m \leq 20$	90

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu theo định dạng sau:

- dòng 1: N
- dòng $2 + k$ ($0 \leq k$): $A[k] B[k]$
- dòng cuối cùng: -1

Mỗi dòng trừ dòng đầu và dòng cuối thể hiện một kịch bản. Kịch bản được ghi trên dòng thứ $2 + k$ là kịch bản thứ k . Trong kịch bản k , túi A chứa $A[k]$ xu và túi B chứa $B[k]$ xu.

Trình chấm mẫu sẽ gọi hàm `devise_strategy(N)`. Giá trị của x là độ dài của mảng trả về trừ đi một. Sau đó, nếu trình chấm mẫu xác định rằng mảng trả về bởi hàm `devise_strategy` không tuân theo các ràng buộc được mô tả trong phần Chi tiết cài đặt, trình chấm mẫu sẽ in ra một trong các lỗi sau và kết thúc:

- `s` is an empty array: s là một mảng rỗng (không phải là một chiến thuật đúng đắn).
- `s[i]` contains incorrect length: Tồn tại một chỉ số i ($0 \leq i \leq x$) mà độ dài của mảng $s[i]$ không phải là $N + 1$.
- First element of `s[i]` is non-binary: Tồn tại một chỉ số i ($0 \leq i \leq x$) mà $s[i][0]$ không phải là 0 hay 1.
- `s[i][j]` contains incorrect value: Tồn tại cặp chỉ số i, j ($0 \leq i \leq x, 1 \leq j \leq N$) mà $s[i][j]$ có giá trị không nằm trong khoảng -2 và x .

Ngược lại, trình chấm mẫu đưa ra hai luồng kết quả đầu ra.

Luồng thứ nhất, trình chấm mẫu in ra kết quả của chiến thuật của bạn theo định dạng như sau:

- dòng $1 + k$ ($0 \leq k$): kết quả của chiến thuật của bạn với kịch bản k . Nếu thực hiện chiến thuật của bạn dẫn đến việc có một tù nhân xác định túi A chứa ít xu hơn, kết quả sẽ là kí tự A. Nếu thực hiện chiến thuật của bạn dẫn đến việc có một tù nhân xác định túi B chứa ít xu hơn, kết quả sẽ là kí tự B. Nếu thực hiện chiến thuật của bạn dẫn tới không có tù nhân nào xác định túi chứa ít xu hơn, kết quả sẽ là kí tự X.

Luồng thứ hai, trình chấm mẫu viết ra một file `log.txt` trong thư mục hiện tại với định dạng như sau:

- dòng $1 + k$ ($0 \leq k$): $w[k][0] w[k][1] \dots$

Dãy số trên dòng thứ $1 + k$ tương ứng với kịch bản k mô tả các số được viết trên bảng. Cụ thể, $w[k][l]$ là số được viết bởi tù nhân thứ $l + 1$.