



## چالش زندان بان

در یک زندان، ۵۰۰ نفر زندانی هستند. یک روز، نگهبان به زندانیان شانس برای آزاد شدن پیشنهاد می‌دهد. او دو کیسه پول با نام‌های کیسه A و کیسه B در یک اتاق قرار می‌دهد که هر کیسه شامل تعدادی (بین ۱ تا  $N$ ) سکه است. تعداد سکه‌ها در دو کیسه با هم متفاوت‌اند. زندان بان این چالش را به زندانیان ارائه می‌کند: کیسه با تعداد سکه کمتر را پیدا کنید.

در اتاق علاوه بر کیسه‌های پول یک وایت‌برد هم قرار دارد. در هر لحظه دقیقاً یک عدد روی وایت‌برد نوشته شده است. عددی که ابتدا روی وایت‌برد نوشته شده 0 است.

نگهبان از زندانیان می‌خواهد که یکی یکی وارد اتاق شوند. هر زندانی که وارد اتاق می‌شود نمی‌داند که قبل از او کدام زندانیان و چه تعداد از آن‌ها وارد اتاق شده‌اند. هر زمان که یک زندانی وارد اتاق می‌شود، عددی که در حال حاضر روی وایت‌برد نوشته شده را می‌خواند. بعد از خواندن عدد، زندانی باید یکی از دو کیسه A و B را انتخاب کند. او سپس کیسه انتخاب شده را بررسی می‌کند تعداد دقیق سکه‌های داخل کیسه را می‌فهمد. سپس، زندانی باید یکی از دو عمل زیر را انجام دهد:

- روی وایت‌برد به جای عدد قبلی یک عدد صحیح غیرمنفی بنویسد و اتاق را ترک کند. عدد جدید می‌تواند متفاوت با عدد قبلی یا برابر آن باشد. بعد از این عمل، چالش ادامه پیدا می‌کند (مگر این که تمام ۵۰۰ زندانی وارد اتاق شده باشند).
- یکی از دو کیسه را به عنوان کیسه با سکه کمتر اعلام کند. با این عمل، چالش بلافاصله خاتمه می‌یابد.

هر زندانی که از اتاق خارج شود، مجدداً توسط نگهبان به اتاق فراخوانده نخواهد شد.

اگر یکی از زندانیان به درستی کیسه شامل سکه کمتر را تشخیص دهد، کل زندانیان برنده خواهند شد. اگر یکی از زندانیان به اشتباه کیسه‌ای را به عنوان کیسه با سکه کمتر اعلام کند، یا این که تمام ۵۰۰ زندانی وارد اتاق شده، ولی هیچ یک کیسه‌ای را به عنوان کیسه با سکه کمتر اعلام نکنند، کل زندانیان بازنده خواهند شد.

پیش از شروع چالش، زندانیان در حال زندان جمع می‌شوند و در مورد **استراتژی** مشترک خود در سه مرحله زیر تصمیم می‌گیرند:

- آن‌ها یک عدد صحیح نامنفی  $x$  را به عنوان بزرگ‌ترین عددی که ممکن است بخواهند روی وایت‌برد بنویسند انتخاب می‌کنند.
- آن‌ها تصمیم می‌گیرند به ازای هر عدد  $i$  که روی وایت‌برد نوشته شده ( $0 \leq i \leq x$ )، کدام کیسه باید توسط زندانی‌ای که عدد  $i$  را هنگام ورود به اتاق از روی تخته می‌خواند بررسی شود.
- آن‌ها تصمیم می‌گیرند که هر زندانی در اتاق پس از فهمیدن تعداد سکه‌ها در کیسه انتخاب شده چه عملی را باید انجام دهد. مشخصاً، به ازای هر عدد  $i$  نوشته شده روی وایت‌برد ( $0 \leq i \leq x$ ) و هر تعداد  $j$  از سکه‌ها که در کیسه بررسی شده دیده شده است ( $1 \leq j \leq N$ )، یکی از دو تصمیم زیر را می‌گیرند:
  - چه عددی بین 0 تا  $x$  باید روی وایت‌برد نوشته شود، یا
  - کدام کیسه باید به عنوان کیسه دارای سکه کمتر اعلام شود.

در صورت برنده شدن چالش، نگهبان زندانیان را پس از  $x$  روز دیگر آزاد می‌کند.

وظیفه شما این است که برای زندانیان یک استراتژی طراحی کنید که مطمئن شوید چالش را (مستقل از تعداد سکه‌های درون کیسه‌های  $A$  و  $B$ ) خواهند برد. امتیاز راه‌حل شما وابسته به مقدار  $x$  خواهد بود. (برای جزئیات بیشتر، بخش زیرمسئله‌ها را ببینید.)

## Implementation Details

You should implement the following procedure:

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

\*  $N$ : the maximum possible number of coins in each bag. \* This procedure should return an array  $s$  of arrays of  $N + 1$  integers, representing your strategy. The value of  $x$  is the length of array  $s$  minus one. For each  $i$  such that  $0 \leq i \leq x$ , the array  $s[i]$  represents what a prisoner should do if they read number  $i$  from the whiteboard upon entering the room:

1. The value of  $s[i][0]$  is 0 if the prisoner should inspect bag A, or 1 if the prisoner should inspect bag B.
2. Let  $j$  be the number of coins seen in the chosen bag. The prisoner should then perform the following action: \* If the value of  $s[i][j]$  is  $-1$ , the prisoner should identify bag A as the one with fewer coins. \* If the value of  $s[i][j]$  is  $-2$ , the prisoner should identify bag B as the one with fewer coins. \* If the value of  $s[i][j]$  is a nonnegative number, the prisoner should write that number on the whiteboard. Note that  $s[i][j]$  must be at most  $x$ . \* This procedure is called exactly once.

## Example

Consider the following call:

```
devise_strategy(3)
```

Let  $v$  denote the number the prisoner reads from the whiteboard upon entering the room. One of the correct strategies is as follows:

- If  $v = 0$  (including the initial number), inspect bag A.
  - If it contains 1 coin, identify bag A as the one with fewer coins.
  - If it contains 3 coins, identify bag B as the one with fewer coins.
  - If it contains 2 coins, write 1 on the whiteboard (overwriting 0).
- If  $v = 1$ , inspect bag B.
  - If it contains 1 coin, identify bag B as the one with fewer coins.
  - If it contains 3 coins, identify bag A as the one with fewer coins.
  - If it contains 2 coins, write 0 on the whiteboard (overwriting 1). Note that this case can never happen, as we can conclude that both bags contain 2 coins, which is not allowed.

To report this strategy the procedure should return  $[[0, -1, 1, -2], [1, -2, 0, -1]]$ . The length of the returned array is 2, so for this return value the value of  $x$  is  $2 - 1 = 1$ .

## Constraints

\*  $2 \leq N \leq 5000$

## Subtasks

1. (5 points)  $N \leq 500$ , the value of  $x$  must not be more than 500.
2. (5 points)  $N \leq 500$ , the value of  $x$  must not be more than 70.
3. (90 points) The value of  $x$  must not be more than 60.

If in any of the test cases, the array returned by `devise_strategy` does not represent a correct strategy, the score of your solution for that subtask will be 0.

In subtask 3 you can obtain a partial score. Let  $m$  be the maximum value of  $x$  for the returned arrays over all test cases in this subtask. Your score for this subtask is calculated according to the following table:

Condition	Points
$40 \leq m \leq 60$	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

## Sample Grader

The sample grader reads the input in the following format:

\* line 1:  $N$  \* line  $2 + k$  ( $0 \leq k$ ):  $A[k] B[k]$  \* last line:  $-1$

Each line except the first and the last one represents a scenario. We refer to the scenario described in line  $2 + k$  as scenario  $k$ . In scenario  $k$  bag A contains  $A[k]$  coins and bag B contains  $B[k]$  coins.

The sample grader first calls `devise_strategy(N)`. The value of  $x$  is the length of the array returned by the call minus one. Then, if the sample grader detects that the array returned by `devise_strategy` does not conform to the constraints described in Implementation Details, it prints one of the following error messages and exits:

\* `s` is an empty array: `s` is an empty array (which does not represent a valid strategy).  
\* `s[i]` contains incorrect length: There exists an index  $i$  ( $0 \leq i \leq x$ ) such that the length of `s[i]` is not  $N + 1$ .  
\* First element of `s[i]` is non-binary: There exists an index  $i$  ( $0 \leq i \leq x$ ) such that `s[i][0]` is neither 0 nor 1.  
\* `s[i][j]` contains incorrect value: There exist indices  $i, j$  ( $0 \leq i \leq x, 1 \leq j \leq N$ ) such that `s[i][j]` is not between  $-2$  and  $x$ .

Otherwise, the sample grader produces two outputs.

First, the sample grader prints the output of your strategy in the following format:

\* line  $1 + k$  ( $0 \leq k$ ): output of your strategy for scenario  $k$ . If applying the strategy leads to a prisoner identifying bag A as the one with fewer coins, then the output is the character A. If applying the strategy leads to a prisoner identifying bag B as the one with fewer coins, then the output is the character B. If applying the strategy does not lead to any prisoner identifying a bag with fewer coins, then the output is the character X.

Second, the sample grader writes a file `log.txt` in the current directory in the following format:

\* line  $1 + k$  ( $0 \leq k$ ): `w[k][0] w[k][1] ...`

The sequence on line  $1 + k$  corresponds to scenario  $k$  and describes the numbers written on the whiteboard. Specifically, `w[k][l]` is the number written by the  $(l + 1)^{th}$  prisoner to enter the room.