



Reto de prisioneros

En una prisión hay 500 prisioneros. Un día, el guardián les ofrece la oportunidad de liberarlos. El guardián coloca dos bolsas con dinero, la bolsa A y la bolsa B, en una habitación. Cada bolsa contiene entre 1 y N monedas, inclusive. El número de monedas en la bolsa A es **diferente** al número de monedas en la bolsa B. El guardián les presenta un reto a los prisioneros. El objetivo de los prisioneros es identificar la bolsa con menos monedas.

La habitación, además de las bolsas de monedas, también contiene una pizarra. En cada momento, hay un único número escrito en la pizarra. Inicialmente, el número en la pizarra es 0.

Luego, el guardián les pide a los prisioneros que entren en la habitación, uno por uno. Cuando un prisionero entra a la habitación no sabe ni cuáles ni cuántos de los otros prisioneros entraron a la habitación antes que él. Cuando un prisionero entra a la habitación, lee el número escrito actualmente en la pizarra. Después de leer el número, debe elegir entre la bolsa A o la bolsa B. Luego, el prisionero **inspecciona** la bolsa elegida, de este modo sabrá el número de monedas dentro de ella.

Después, el prisionero debe realizar una de las siguientes dos **acciones**:

- Sobrecribir el número en la pizarra con un número entero no negativo y salir de la habitación. El prisionero puede cambiar o mantener el número actual. El reto continúa después de eso (a menos que todos los 500 prisioneros hayan entrado a la habitación).
- Identificar una bolsa como la que contiene menos cantidad de monedas. Esto termina el reto inmediatamente.

El guardián nunca le pedirá a un prisionero que ya haya salido de la habitación, que ingrese nuevamente en ella.

Los prisioneros ganan el reto si uno de ellos identifica correctamente la bolsa con menos monedas. Pierden si cualquiera de ellos identifica la bolsa incorrectamente, o si todos los 500 prisioneros pasan por la habitación sin haber intentado identificar la bolsa con menos monedas.

Antes de que el reto comience, los prisioneros se reúnen en el corredor de la prisión y deciden una **estrategia** en común para el reto en tres pasos.

- Eligen un entero no negativo x , que será el número más grande que podrán escribir en la pizarra.

- Deciden, para cada número i escrito en la pizarra ($0 \leq i \leq x$), cual bolsa debe ser inspeccionada por un prisionero que lea el número i en la pizarra al entrar a la habitación.
- Deciden la acción que un prisionero que esté en la habitación debe realizar luego de saber el número de monedas en la bolsa elegida. Específicamente, para cada número i escrito en la pizarra ($0 \leq i \leq x$) y cada número de monedas j vistas en la bolsa inspeccionada ($1 \leq j \leq N$), deben elegir:
 - o bien qué número entre 0 y x (inclusive) debe ser escrito en la pizarra,
 - o bien cuál bolsa debe ser identificada como la que tiene menos monedas.

Al ganar el reto, el guardián liberará a los prisioneros luego de pasar x días más en la prisión.

Tu tarea es idear una estrategia para que los prisioneros se aseguren ganar el reto (sin importar el número de monedas en la bolsa A y la bolsa B). El puntaje de tu solución depende del valor de x (ver la sección de Subtareas para más detalles).

Detalles de implementación

Debes implementar la siguiente función:

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

- N : el máximo número posible de monedas en cada bolsa.
- La función debe retornar un arreglo s de arreglos de $N + 1$ enteros, que representa la estrategia. El valor de x es la longitud del arreglo s menos uno. Para cada i tal que $0 \leq i \leq x$, el arreglo $s[i]$ representa lo que el prisionero debe hacer si lee el número i en la pizarra al entrar a la habitación:
 1. El valor de $s[i][0]$ es 0 si el prisionero debe inspeccionar la bolsa A, o 1 si el prisionero debe inspeccionar la bolsa B.
 2. Sea j el número de monedas vistas en la bolsa elegida. Entonces el prisionero debe realizar la siguiente acción:
 - Si el valor de $s[i][j]$ es -1 , el prisionero debe identificar la bolsa A como la que tiene menos monedas.
 - Si el valor de $s[i][j]$ es -2 , el prisionero debe identificar la bolsa B como la que tiene menos monedas.
 - Si el valor de $s[i][j]$ es un número no negativo, el prisionero debe escribir ese número en la pizarra. Notar que $s[i][j]$ debe ser a lo sumo x .
- Esta función se llama exactamente una vez.

Ejemplo

Considera la siguiente llamada:

```
devise_strategy(3)
```

Sea v el número que el prisionero lee de la pizarra al entrar a la habitación. Una de las estrategias correctas es la siguiente:

- Si $v = 0$ (incluyendo el número inicial), inspecciona la bolsa A.
 - Si contiene 1 moneda, identifica la bolsa A como la que tiene menos monedas.
 - Si contiene 3 monedas, identifica la bolsa B como la que tiene menos monedas.
 - Si contiene 2 monedas, escribe 1 en la pizarra (sobreescribiendo 0).
- Si $v = 1$, inspecciona bolsa B.
 - Si contiene 1 moneda, identifica la bolsa B como la que tiene menos monedas.
 - Si contiene 3 monedas, identifica la bolsa A como la que tiene menos monedas.
 - Si contiene 2 monedas, escribe 0 en la pizarra (sobreescribiendo 1). Ten en cuenta que este caso nunca puede pasar, ya que concluiríamos que ambas bolsas tienen 2 monedas, lo cual no está permitido.

Para reportar esta estrategia esta función debe retornar $[[0, -1, 1, -2], [1, -2, 0, -1]]$. La longitud del arreglo retornado es 2, entonces para este valor retornado el valor de x es $2 - 1 = 1$.

Restricciones

- $2 \leq N \leq 5000$

Subtareas

1. (5 puntos) $N \leq 500$, el valor de x no debe ser más de 500.
2. (5 puntos) $N \leq 500$, el valor de x no debe ser más de 70.
3. (90 puntos) El valor de x no debe ser más de 60.

Si en cualquiera de los casos de prueba, el arreglo retornado por `devise_strategy` no representa una estrategia correcta, el puntaje de tu solución será 0.

En la subtarea 3 puedes obtener un puntaje parcial. Sea m el máximo valor de x para los arreglos retornados para todos los casos de prueba en esta subtarea. Tu puntaje para esta subtarea se calcula de acuerdo a la siguiente tabla:

Condición	Puntos
$40 \leq m \leq 60$	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

Evaluador local

El evaluador local lee la entrada con el siguiente formato:

- línea 1: N
- línea $2 + k$ ($0 \leq k$): $A[k]$ $B[k]$
- última línea: -1

Cada línea excepto la primera y la última representa un escenario. Nos referimos al escenario descrito en la línea $2 + k$ como el escenario k . En el escenario k la bolsa A contiene $A[k]$ monedas y la bolsa B contiene $B[k]$ monedas.

El evaluador local primero llama `devise_strategy(N)`. El valor de x es la longitud del arreglo retornado por la llamada menos uno. Por lo tanto, si el evaluador local detecta que el arreglo retornado por `devise_strategy` no se ajusta a las restricciones descritas en los Detalles de implementación, imprime uno de los siguientes mensajes de error y termina:

- `s` is an empty array: s es un arreglo vacío (lo cual no representa una estrategia válida).
- `s[i]` contains incorrect length: Existe un índice i ($0 \leq i \leq x$) tal que la longitud de $s[i]$ no es $N + 1$.
- First element of `s[i]` is non-binary: Existe un índice i ($0 \leq i \leq x$) tal que $s[i][0]$ no es 0 ni 1.
- `s[i][j]` contains incorrect value: Existen índices i, j ($0 \leq i \leq x, 1 \leq j \leq N$) tal que $s[i][j]$ no está entre -2 y x .

En caso contrario, el evaluador local produce dos salidas.

Primero, el evaluador local imprime la salida de tu estrategia en el siguiente formato:

- línea $1 + k$ ($0 \leq k$): salida de tu estrategia para el escenario k . Si la aplicación de la estrategia conduce a un prisionero a identificar la bolsa A como la que tiene menos monedas, entonces

la salida es el caracter A. Si la aplicación de la estrategia conduce a un prisionero a identificar la bolsa B como la que tiene menos monedas, entonces la salida es el caracter B. Si la aplicación de la estrategia no conduce a ningún prisionero a identificar la bolsa con menos monedas, entonces la salida es el caracter X.

Segundo, el evaluador local escribe un archivo `log.txt` en el directorio actual en el siguiente formato:

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

La secuencia en la línea $1 + k$ corresponde al escenario k y describe los números escritos en la pizarra. Específicamente, $w[k][l]$ es el número escrito por el $(l + 1)$ -ésimo prisionero en entrar a la habitación.