

## Organizando el Librero

Contribución de Gastón Fontenla Núñez

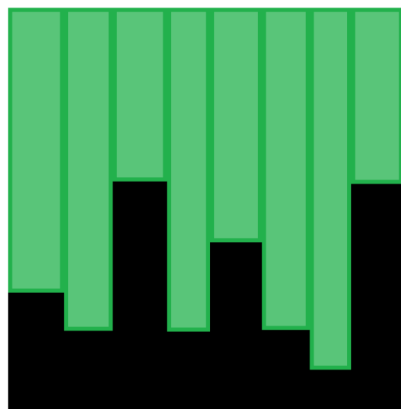
### Descripción del problema

Dimitri es un niño fanático de la literatura rusa. Hoy es su cumpleaños y sus familiares y amigos le obsequiaron  $N$  libros de famosísimos autores rusos. Lleno de felicidad, Dimitri llevó todos sus libros hasta su cuarto, y para su sorpresa, le habían regalado también un nuevo librero. Notó que éste tenía bases de diversas alturas, y que en cada base cabía un solo libro. Había tantas bases como libros.

Al observar esto, Dimitri decidió que solo podría quedarse con los libros si era posible lograr que todos quedasen alineados a la misma altura. Es decir, una vez colocados (en algún orden) todos los libros tienen que coincidir en su parte superior. Los libros, al igual que las bases, vienen en varios tamaños, aunque solo nos interesa la altura.

Mientras Dimitri disfruta del resto de su día, tu tarea es determinar el orden (de izquierda a derecha) en el que irán colocados los libros, de tal forma que las alturas finales (libro + base) sean iguales.

Además, debes calcular de cuántas formas se puede lograr un ordenamiento que cumpla con lo pedido. Dos ordenamientos  $A$  y  $B$  son distintos si existe al menos una posición  $i$  tal que  $A_i \neq B_i$ .



En negro, las bases. En verde, los libros.

### Detalles de implementación

Debes implementar la función `librero(bases, libros, orden)`, que dados los arreglos de enteros `bases` y `libros` (ambos de tamaño  $N$ ), retorne un arreglo con dos elementos:

- El primer entero del arreglo retornado será la altura final resultante.
- El segundo entero del arreglo será la cantidad de ordenamientos existentes (**módulo 1.000.000.007**)

Además, la función debe calcular un ordenamiento posible de los libros para obtener la altura deseada, que debe almacenarse en el parámetro `orden`, utilizando números de 1 a  $N$  sin repetir (1 es el primer libro en el orden de la entrada,  $N$  es el último libro en el orden de la entrada). En caso de existir más de un ordenamiento posible que cumpla estas características, cualquiera vale.

En el caso de que no sea posible cumplir con el requisito solicitado de que todos los libros queden a una misma altura total en el librero, se debe retornar una altura de  $-1$ .

**Evaluador local**

El evaluador local lee de la entrada estándar:

- Una línea con la cantidad de libros  $N$
- Una segunda línea con  $N$  enteros, que representan las alturas de las distintas bases, en orden.
- Una tercera línea con  $N$  enteros, que representan las alturas de los distintos libros, en el orden en que estos se numeran desde 1 hasta  $N$ .

Devuelve por la salida estándar una línea con el resultado de llamar a la función `librero`, y una segunda línea con el ordenamiento de libros calculado por la función.

**Cotas**

- $1 \leq N \leq 100.000$
- $1 \leq \text{bases}[i], \text{libros}[i] \leq 1.000.000.000$

**Ejemplos**

Si el evaluador local recibe la siguiente entrada:

```
8
3 2 6 2 4 2 1 6
6 8 4 8 7 9 8 4
```

Una implementación correcta podrá devolver:

```
10 12
5 2 3 4 1 7 6 8
```

Si en cambio recibe:

```
4
1 1 1 21
4 4 4 4
```

Para una implementación correcta devolverá:

```
-1 0
```

**Puntajes**

Se recibe el 20% del puntaje por calcular correctamente la altura total, 40% del puntaje por calcular correctamente la cantidad de ordenamientos que cumplen lo pedido, y el 40% restante si además de calcular correctamente la altura, el orden propuesto para los libros es correcto.

**Subtareas**

1.  $N = 2$  (5 puntos)
2.  $N = 3$  (8 puntos)
3. Se puede cumplir con el requisito solicitado de que todos los libros queden a una misma altura total en el librero (12 puntos)
4.  $N \leq 10$  (25 puntos)
5.  $N \leq 10.000$  (25 puntos)
6.  $N \leq 100.000$  (25 puntos)