



## Balanceos

Amina tiene 6 monedas numeradas del **1** al **6**, ella sabe que sus monedas tienen diferentes pesos y le gustaría ordenarlas de acuerdo a su peso, es por eso que ha desarrollado un nuevo tipo de balanza.

Una balanza tradicional tiene dos platos, para usarla, se coloca una moneda en cada uno de los platos y la balanza determina qué moneda es más pesada.

La nueva balanza de Amina es más compleja, tiene 4 platos etiquetados **A**, **B**, **C** y **D**. Además tiene 4 diferentes configuraciones, cada una de ellas responde diferentes preguntas respecto a las monedas. Para usar la balanza, Amina debe colocar exactamente una moneda en cada uno de los platos **A**, **B**, y **C**. Adicionalmente, en la cuarta configuración debe colocar también exactamente una moneda en el plato **D**.

Las cuatro configuraciones instruirán a la balanza para responder las siguientes 4 preguntas:

1. ¿Cuál de las monedas en los platos **A**, **B**, y **C** es la más pesada?
2. ¿Cuál de las monedas en los platos **A**, **B**, y **C** es la más liviana?
3. ¿Cuál de las monedas en los platos **A**, **B**, y **C** es la mediana? (Es decir la moneda que no es ni la más pesada ni la más liviana de las tres.)
4. De las monedas en los platos **A**, **B** y **C**, considera solo las monedas que son más pesadas que la moneda en el plato **D**, si existe al menos una de ellas, ¿cuál de ellas es la más liviana? De otro modo, si no hay de esas monedas, ¿cuál de las monedas en los platos **A**, **B**, y **C** es la más liviana?

## Problema

Escribe un programa que ordene las seis monedas de Amina de acuerdo a su peso. Tu programa puede hacer preguntas a la balanza de Amina para comparar los pesos de las monedas. Varios casos de prueba serán dados a tu programa para que los resuelva, cada uno de ellos correspondiendo a un nuevo conjunto de seis monedas.

Tu programa deberá implementar las funciones `init` y `orderCoins`. Durante cada ejecución de tu programa, el evaluador llamará en primer lugar a la función `init` exactamente una vez. Esto te dará el número de casos de prueba y te permitirá inicializar cualquier variable. Después el evaluador llamará la función `orderCoins()` una vez por cada caso de prueba.

- `init(T)`
  - `T`: El número de casos de prueba que tu programa deberá resolver durante la ejecución actual. `T` es un entero dentro del rango **1, ..., 18**.
  - Esta función no regresa ningún valor.

- `orderCoins()`
  - Esta función será llamada exactamente una vez por cada caso de prueba.
  - La función debe determinar el orden correcto de las monedas de Amina haciendo uso de las funciones del evaluador `getHeaviest()`, `getLightest()`, `getMedian()`, y/o `getNextLightest()`.
  - Una vez que la función conoce el orden correcto, deberá reportarlo llamando a la función del evaluador `answer()`.
  - Después de ejecutar `answer()`, la función `orderCoins()` deberá terminar, esta función no regresa ningún valor.

Puedes usar las siguientes funciones del evaluador en tu programa:

- `answer(W)` — tu programa deberá usar esta función para reportar la solución que haya encontrado.
  - `W`: Un arreglo de 6 elementos conteniendo el orden correcto de las monedas. `W[0]` hasta `W[5]` deberán contener los números de las monedas (es decir números de **1** a **6**) ordenados desde la más liviana hasta la más pesada.
  - Tu programa deberá llamar esta función solo una vez por cada caso de prueba.
  - Esta función no regresa ningún valor.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — estas funciones corresponden a las configuraciones 1, 2 y 3 de la balanza de Amina respectivamente.
  - `A, B, C`: Las monedas colocadas en los platos **A**, **B**, y **C**, respectivamente. **A**, **B** y **C** deben ser tres enteros distintos, cada uno de ellos entre **1** y **6** inclusive.
  - Cada una de estas funciones regresa uno de los números `A`, `B`, o `C`: El número de la moneda apropiada, por ejemplo, `getHeaviest(A, B, C)` regresa el número de la moneda más pesada de las tres monedas dadas
- `getNextLightest(A, B, C, D)` — esta función corresponde a la configuración 4 de la balanza de Amina.
  - `A, B, C, D`: Las monedas colocadas en los platos **A**, **B**, **C**, y **D**, respectivamente. `A`, `B`, `C`, y `D` deben ser cuatro enteros positivos distintos, cada uno de ellos entre **1** y **6** inclusive.
  - La función regresa uno de los números `A`, `B`, o `C`: El número de la moneda seleccionada por la balanza como se describe arriba para la configuración 4. Esto es, la moneda regresada es la más liviana de las monedas en los platos **A**, **B**, y **C** que son más pesadas que la moneda en el plato **D**; o, si ninguna de ellas es más pesada que la moneda en el plato **D**, el valor regresado es el de la moneda más liviana de las tres monedas colocadas en los platos **A**, **B**, y **C**.

## Evaluación

No hay subtarear en este problema. En vez de eso, tu puntaje se basará en cuantos balanceos (número total de llamadas las funciones del evaluador `getLightest()`, `getHeaviest()`, `getMedian()` y/o `getNextLightest()`) tu programa hace.

Tu programa será ejecutado varias veces con múltiples casos de prueba en cada ejecución. Considera  $r$  como el número de ejecuciones de tu programa, este número es determinado por los casos de prueba. Si tu programa no ordena las monedas correctamente en cualquier caso de prueba de cualquier ejecución, será evaluado con 0 puntos. De otra forma, las ejecuciones serán evaluadas individualmente como se describe a continuación.

Considera  $Q$  como el menor número de modo que es posible ordenar cualquier secuencia de seis monedas usando  $Q$  balanceos de la balanza de Amina. Para hacer la tarea más interesante, no revelaremos el valor de  $Q$  aquí.

Supón que el número más grande de balanceos a lo largo de todos los casos de prueba de todas las ejecuciones es  $Q + y$  para algún entero  $y$ . Después considera una sola ejecución de tu programa, toma el número más grande de balanceos a lo largo de todos los  $T$  casos de prueba en dicha ejecución como  $Q + x$  para un entero no negativo  $x$ . (Si usas menos de  $Q$  balanceos para cada caso de prueba, entonces  $x = 0$ .) Entonces, el puntaje de dicha ejecución será  $\frac{100}{r((x+y)/5+1)}$ , truncado a dos dígitos después del punto decimal.

En particular, si tu programa hace a lo más  $Q$  balanceos para cada caso de prueba de cada ejecución, obtendrás 100 puntos.

## Ejemplo

Supón que las monedas están ordenadas **3 4 6 2 1 5** de la más liviana a la más pesada.

Llamada a función	Valor de retorno	Explicación
getMedian(4, 5, 6)	6	La moneda <b>6</b> es la mediana entre las monedas <b>4</b> , <b>5</b> , y <b>6</b> .
getHeaviest(3, 1, 2)	1	La moneda <b>1</b> es la más pesada entre las monedas <b>1</b> , <b>2</b> , y <b>3</b> .
getNextLightest(2, 3, 4, 5)	3	Las monedas <b>2</b> , <b>3</b> , y <b>4</b> son todas más livianas que la moneda <b>5</b> , entonces la más liviana de ellas ( <b>3</b> ) es regresada.
getNextLightest(1, 6, 3, 4)	6	Las monedas <b>1</b> y <b>6</b> son más pesadas que la moneda <b>4</b> . Entre las monedas <b>1</b> y <b>6</b> , la moneda <b>6</b> es la más liviana.
getHeaviest(3, 5, 6)	5	La moneda <b>5</b> es la más pesada entre las monedas <b>3</b> , <b>5</b> y <b>6</b> .
getMedian(1, 5, 6)	1	La moneda <b>1</b> es la mediana entre las monedas <b>1</b> , <b>5</b> y <b>6</b> .
getMedian(2, 4, 6)	6	La moneda <b>6</b> es la mediana entre las monedas <b>2</b> , <b>4</b> y <b>6</b> .
answer([3, 4, 6, 2, 1, 5])		El programa encontró la respuesta correcta para este caso de prueba.

## Evaluador de ejemplo

El evaluador de ejemplo lee las entradas en el siguiente formato:

- Línea **1**:  $T$  — el número de casos de prueba
- Cada una de las líneas desde **2** hasta  $T + 1$ : una secuencia de **6** distintos números de **1** a **6** inclusive: el orden de las monedas de la más liviana a la más pesada.

Por ejemplo, una entrada que consiste de dos casos de prueba donde las monedas están ordenadas

**1 2 3 4 5 6** y **3 4 6 2 1 5** luce de la siguiente forma:

```
2
1 2 3 4 5 6
3 4 6 2 1 5
```

El evaluador de ejemplo imprime el arreglo que es pasado como parámetro a la función `answer()`.