El Gran Premio

El Gran Premio es un juego famoso de TV. Usted es el afortunado concursante que ha avanzado hasta la ronda final. Está parado frente a una fila de n cajas, numeradas del 0 al $n-1$ de izquierda a derecha. Cada caja contiene un premio que no puede ser visto sino hasta que la caja sea abierta. Existen $v \geq 2$ *tipos* diferentes de premios. Los tipos son numerados del 1 al v en orden *descendente* de valor.

El premio de tipo 1 es el mas valioso: un diamante. Existe exactamente un diamante en las cajas. El premio de tipo v es el menos valioso: una paleta. Para hacer el juego más emocionante, el número de premios de bajo valor es mucho mayor que el número de premios más valiosos. Más concretamente, para todo t tal que $2 \leq t \leq v$ sabemos lo siguiente: si existen k premios de tipo $t-1$, existen *estrictamente más* de k^2 premios de tipo t.

Su objetivo es ganar el diamante. Al final del juego usted tendrá que abrir una caja y recibirá el premio que esta contiene. Antes de elegir qué caja abrir, puede hacerle a Rambod, el presentador del juego, algunas preguntas. Para cada pregunta, usted elige alguna caja i. Como respuesta, Rambod le dará un arreglo a conteniendo dos enteros.

Su significado es el siguiente:

- Entre las cajas a la izquierda de la caja i hay exactamente $a[0]$ cajas que contienen un premio más valioso que el de la caja i.
- Entre las cajas a la derecha de la caja i hay exactamente $a[1]$ cajas que contienen premios más valiosos que el de la caja i.

Por ejemplo, suponga que $n = 8$. En su pregunta, usted elige la caja $i = 2$. Rambold le responde $a = [1, 2]$. Esta respuesta significa que:

- Exactamente una de las cajas 0 y 1 contienen un premio más valioso que el de la caja 2.
- Exactamente dos de las cajas $3, 4, \ldots, 7$ contienen un premio más valioso que el de la caja 2.

Su tarea es encontrar la caja que contiene el diamante haciendo la menor cantidad de preguntas posible.

Detalles de implementación

Debe implementar el siguiente procedimiento:

```
int find_best(int n)
```

- Este procedimiento es llamado exactamente una vez por el calificador.
- n: el número de cajas.
- El procedimiento debe retornar el número de la caja que contiene el diamante. Es decir, el único entero d $(0 \leq d \leq n-1)$ tal que la caja d contiene un premio de tipo 1.

El procedimiento anterior puede hacer llamadas al siguiente procedimiento:

```
int[] ask(int i)
```

- i: el número de la caja por la que decide preguntar. El valor de i debe estar entre 0 y $n-1$, inclusive.
- Este procedimiento retorna un arreglo a con 2 elementos. Aquí, $a[0]$ es el número de premios más valiosos en las cajas a la izquierda de la caja i y $a[1]$ es el número de premios más valiosos en las cajas a la derecha de la caja i.

Ejemplo

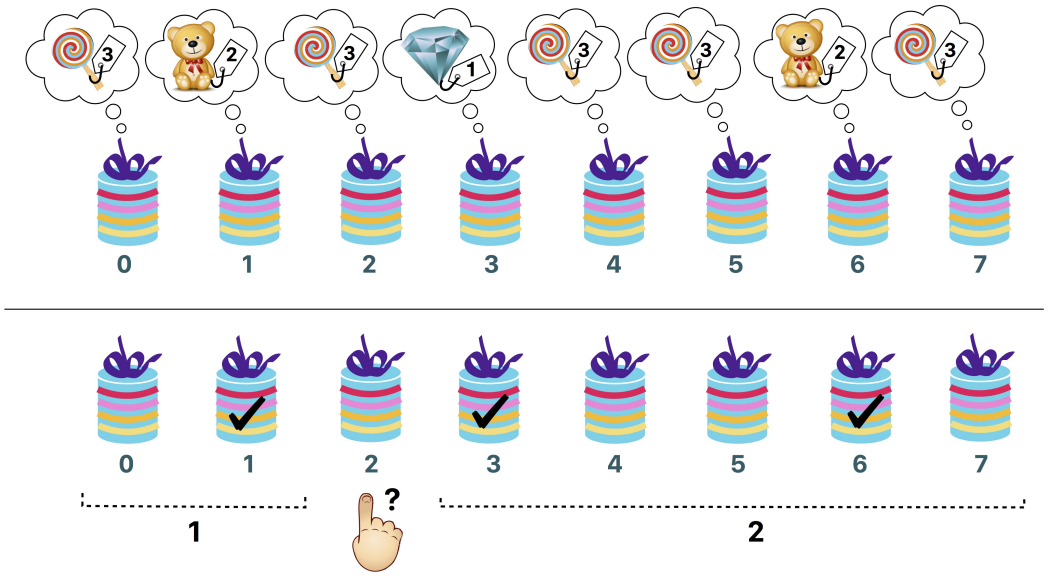
El calificador hace la siguiente llamada:

```
find_best(8)
```

Hay $n = 8$ cajas. Suponga que los tipos de premios son $[3, 2, 3, 1, 3, 3, 2, 3]$. Todos las llamadas posibles al procedimiento `ask` y sus correspondientes valores de retorno son listados a continuación.

- `ask(0)` retorna $[0, 3]$
- `ask(1)` retorna $[0, 1]$
- `ask(2)` retorna $[1, 2]$
- `ask(3)` retorna $[0, 0]$
- `ask(4)` retorna $[2, 1]$
- `ask(5)` retorna $[2, 1]$
- `ask(6)` retorna $[1, 0]$
- `ask(7)` retorna $[3, 0]$

En este ejemplo, el diamante se encuentra en la caja 3. Por lo tanto, el procedimiento `find_best` debe retornar 3.



La figura de arriba ilustra este ejemplo. La parte superior muestra los tipos de los premios en cada caja. La parte inferior ilustra la pregunta `ask(2)`. Las cajas marcadas contienen premios más valiosos que el de caja 2.

Restricciones

- $3 \leq n \leq 200\,000$.
- El tipo del premio en cada caja se encuentra entre 1 y v, inclusive.
- Existe exactamente un premio de tipo 1.
- Para todo $2 \leq t \leq v$, si existen k premios de tipo $t-1$, existen *estrictamente* más de k^2 premios de tipo t.

Subtareas y puntuación

En algunos casos de prueba el comportamiento del calificador es adaptable. Esto significa que en estos casos de prueba el calificador no tiene una secuencia fija de premios. En su lugar, las respuestas devueltas por el calificador pueden depender de las preguntas hechas por su solución. Se garantiza que el calificador responde de tal manera que después de cada pregunta existe al menos una secuencia de premios consistente con todas las respuestas devueltas hasta ese momento.

1. (20 puntos) Existe exactamente 1 diamante y $n-1$ paletas (por lo tanto, $v = 2$). Puede llamar al procedimiento `ask` a lo sumo $10\,000$ veces.
2. (80 puntos) Sin restricciones adicionales.

En la subtarea 2 puede obtener una puntuación parcial. Sea q el máximo número de llamadas al procedimiento `ask` entre todos los casos de prueba en esta subtarea. Entonces, su puntuación para esta subtarea se calcula de acuerdo a la siguiente tabla:

Preguntas	Puntuación
$10\,000 < q$	0 (reportada en CMS como 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Calificador de ejemplo

El calificador de ejemplo no es adaptable. En su lugar, únicamente lee y usa un arreglo fijo p de tipos de premios. Para todo $0 \leq b \leq n - 1$, el tipo del premio en la caja b es dado como $p[b]$. El calificador de ejemplo espera entradas en el siguiente formato:

- línea 1: n
- línea 2: $p[0]$ $p[1]$ $\ldots$ $p[n-1]$

El calificador de ejemplo imprime una sóla línea que contiene el valor de retorno de `find_best` y el número de llamadas al procedimiento `ask`.