

# Empacando Bizcochos (biscuits)

La tía Khong está organizando una competencia con  $x$  participantes, y quiere dar a cada participante una **bolsa de bizcochos**. Hay " $k$ " tipos diferentes de bizcochos, numerados de 0 a  $k - 1$ . Cada bizcocho del tipo  $i$  ( $0 \leq i \leq k - 1$ ) tiene un **nivel de sabor** de  $2^i$ . La tía Khong tiene  $a[i]$  (posiblemente cero) bizcochos del tipo  $i$  en su despensa.

Cada una de las bolsas de la tía Khong contienen cero o más bizcochos de cada tipo. El número total de bizcochos del tipo  $i$  en todas las bolsas no debe exceder  $a[i]$ . La suma de los niveles de sabor de todos los bizcochos en la bolsa es llamado el **nivel total de sabor** de la bolsa.

Ayuda a la tía Khong a encontrar cuántos valores diferentes de  $y$  existen, tal que sea posible empacar  $x$  bolsas de bizcochos, cada una teniendo un nivel total de sabor de  $y$ .

## Detalles de Implementación

Debes implementar el siguiente procedimiento:

```
int64 count_tastiness(int64 x, int64[] a)
```

- $x$ : el número de bolsas de bizcochos a empacar.
- $a$ : un arreglo de tamaño  $k$ . Para  $0 \leq i \leq k - 1$ ,  $a[i]$  que denota el número de bizcochos del tipo  $i$  en la despensa.
- El procedimiento debe retornar el número de valores diferentes de  $y$ , tal que la tía pueda empacar  $x$  bolsas de bizcochos, cada una teniendo un nivel total de sabor de  $y$ .
- El procedimiento es llamado un total de  $q$  veces (vea las secciones de Límites y Subtareas para los valores permitidos de  $q$ ). Cada una de las llamadas debe ser tratada como un escenario separado.

## Ejemplos

### Ejemplo 1

Considere la siguiente llamada:

```
count_tastiness(3, [5, 2, 1])
```

Esto significa que la tía quiere empacar 3 bolsas, y hay 3 tipos de bizcochos en la despensa:

- 5 bizcochos del tipo 0, cada uno teniendo un nivel de sabor de 1,
- 2 bizcochos del tipo 1, cada uno teniendo un nivel de sabor de 2,
- 1 bizcocho del tipo 2, teniendo un nivel de sabor de 4.

Los valores posibles de  $y$  son  $[0, 1, 2, 3, 4]$ . Por ejemplo, para empacar 3 bolsas con nivel total de sabor de 3, la tía puede empacar:

- una bolsa conteniendo tres bizcochos del tipo 0, y
- dos bolsas, cada una conteniendo un bizcocho del tipo 0 y un bizcocho del tipo 1.

Puesto que hay 5 posibles valores de  $y$ , el procedimiento debe retornar 5.

<div> <div></div> <div></div> <div></div> </div> $y = 0$	<div> <div>1</div> <div>1</div> <div>1</div> </div> $y = 1$	<div> <div>1,1</div> <div>2</div> <div>2</div> </div> $y = 2$
<div> <div>1,1,1</div> <div>1,2</div> <div>1,2</div> </div> $y = 3$	<div> <div>1,1,2</div> <div>1,1,2</div> <div>4</div> </div> $y = 4$	

## Ejemplo 2

Considere la siguiente llamada:

```
count_tastiness(2, [2, 1, 2])
```

Esto significa que la tía quiere empacar 2 bolsas, y existen 3 tipos de bizcochos en la despensa:

- 2 bizcochos del tipo 0, cada uno teniendo un nivel de sabor de 1,
- 1 bizcocho del tipo 1, teniendo un nivel de sabor de 2,
- 2 bizcochos del tipo 2, cada uno teniendo un nivel de sabor de 4.

Los valores posibles de  $y$  son  $[0, 1, 2, 4, 5, 6]$ . Tal que existen 6 posibles valores de  $y$ , el procedimiento debe retornar 6.

## Límites

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$  (para todo  $0 \leq i \leq k - 1$ )

- Para cada llamada a `count_tastiness`, la suma de los niveles de sabor de todos los bizcochos en la despensa no excede  $10^{18}$ .

## Subtareas

1. (9 puntos)  $q \leq 10$ , y para cada llamada a `count_tastiness`, la suma de los niveles de sabor de todos los bizcochos en la despensa no exceden 100 000.
2. (12 puntos)  $x = 1, q \leq 10$
3. (21 puntos)  $x \leq 10\,000, q \leq 10$
4. (35 puntos) El valor de retorno de cada llamada a `count_tastiness` no excede 200 000.
5. (23 puntos) Sin restricciones adicionales.

## Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato. La primera línea contiene un entero  $q$ . Después de esto,  $q$  pares de líneas siguen, y cada par describe un escenario simple en el siguiente formato:

- línea 1:  $k \ x$
- línea 2:  $a[0] \ a[1] \ \dots \ a[k-1]$

La salida del evaluador de ejemplo tiene el siguiente formato:

- línea  $i$  ( $1 \leq i \leq q$ ): retorna el valor de `count_tastiness` para el  $i$ -ésimo escenario en la entrada.