

# Empacando galletas (biscuits)

La tía Khong está organizando una competencia con  $x$  participantes, y quiere darle a cada participante una **bolsa de galletas**. Hay  $k$  tipos diferentes de galletas, numeradas desde 0 hasta  $k - 1$ . Cada galleta del tipo  $i$  ( $0 \leq i \leq k - 1$ ) tiene un **valor de sabor** de  $2^i$ . La tía Khong tiene  $a[i]$  (posiblemente cero) galletas del tipo  $i$  en su despensa.

Cada bolsa de la tía Khong deberá contener cero o más galletas de cada tipo. El número total de galletas del tipo  $i$  en todas las bolsas no deben exceder  $a[i]$ . La suma de todos los sabores de las galletas de una bolsa es llamada el **sabor total** de la bolsa.

Ayuda a la tía Khong a encontrar cuantos valores diferentes de  $y$  existen, de modo que sea posible empacar  $x$  bolsas de galletas, cada una teniendo un sabor total igual a  $y$ .

## Detalles de Implementación

Debes implementar el siguiente procedimiento:

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

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

## Ejemplos

### Ejemplo 1

Considera la siguiente llamada:

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

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

- 5 galletas del tipo 0, cada una teniendo un valor de sabor de 1,
- 2 galletas del tipo 1, cada una teniendo un valor de sabor de 2,
- 1 galleta del tipo 2, teniendo un valor de sabor de 4.

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

- una bolsa conteniendo tres galletes del tipo 0, y
- dos bolsas, cada una conteniendo una galleta del tipo 0 y una del tipo 1.

Dado 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

Considera la siguiente llamada:

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

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

- 2 galletas del tipo 0, cada una teniendo un valor de sabor de 1,
- 1 galleta del tipo 1, teniendo un valor de sabor de 2,
- 2 galletas del tipo 2, cada una teniendo un valor de sabor de 4.

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

## Restricciones

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

- Para cada llamada a `count_tastiness`, la suma de los valores de sabores de todas las galletas 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 valores de sabores de todas las galletas en la despensa no excede 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 correcto retornado de la 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 eso, siguen  $q$  pares de líneas, y cada par de líneas describe un solo escenario con 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  $i$  ( $1 \leq i \leq q$ ): el valor de retorno de `count_tastiness` del  $i$ -ésimo escenario de la entrada.