

Empacando biscochos (biscuits)

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

Cada una de las bolsas de la tía Khong contendrá cero o más biscochos de cada tipo. El número total de biscochos del tipo i en todas las bolsas no debe exceder de $a[i]$. La suma de los valores de sabrosura de todos los biscochos en una bolsa es llamado **total de sabrosura** de la bolsa.

Ayuda a la tía Khong a averiguar cuántos valores y diferentes existen, de tal manera que sea posible empacar x bolsas de biscochos,, cada una de las cuales tiene una sabrosura total igual a y .

Detalles de la implementación

Debería implementar el siguiente procedimiento:

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

- x : el número de bolsas de biscochos a empacar.
- a : au arreglo de longitud k . Para $0 \leq i \leq k - 1$, $a[i]$ denota el número de biscochos de tipo i en la despensa.
- El procedimiento debe retornar el número y de diferentes valores, de tal manera que la tía pueda empaquetar x bolsas de biscochos, cada una con un sabor total de y .
- El procedimiento es llamado un total de q veces (ver las secciones de Restricciones y Subtareas para los valores permitidos de q). Cada una de estas 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 biscochos en la despensa:

- 5 biscochos de tipo 0, cada una con un valor de sabor 1,
- 2 biscochos de tipo 1, cada una con un valor de sabor 2,
- 1 biscochos de tipo 2, con un valor de sabor de 4.

Los posibles valores de y son $[0, 1, 2, 3, 4]$. Por ejemplo, con el objetivo de empacar 3 bolsas con un valor total de sabrosura de 3, la tía puede empacar:

- una bolsa con tres biscochos del tipo 0, y
- dos bolsas, cada una con un biscocho de tipo 0 y un biscocho de tipo 1.

Ya que hay 5 valores posibles de y , el procedimiento debería retornar 5.

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

Ejemplo 2

Considere la siguiente llamada:

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

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

- 2 biscochos de tipo 0, cada una con un valor de sabor 1,
- 1 biscocho de tipo 1, con un valor de sabor de 2,
- 2 biscochos de tipo 2, cada una con un valor de sabor 4.

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

Restricciones

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

- Para cada llamada a `count_tastiness`, la suma de los valores de sabrosura de todos los bizcochos de la despensa no excede a 10^{18} .

Subtareas

1. (9 puntos) $q \leq 10$, y por cada llamada a `count_tastiness`, la suma de los valores de sabrosura de todos los bizcochos de la despensa no excede a 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 de cada llamada a `count_tastiness` no excede a 200 000.
5. (23 puntos) No hay restricciones adicionales.

Grader de ejmplos

El grader 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 describe un solo escenario en el siguiente formato:

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

La salida del grader de ejemplo tiene el siguiente formato:

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