

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 diferentes tipos de galletas, numeradas del 0 a $k - 1$. Cada galleta de 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 de tipo i en su alacena.

Cada una de las bolsas de la tía Khong tendrá cero o más galletas de cada tipo. El número total de galletas de tipo i entre todas las bolsas, no debe exceder $a[i]$. La suma de los valores de sabor en una bolsa se llama **sabor total** de la bolsa.

Ayuda a la tía Khong a averiguar cuántos valores diferentes de y existen, tal que es posible empacar x bolsas de galletas con un sabor total de y para cada bolsa.

Detalles de implementación

Debes implementar la siguiente función:

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

- x : el número de bolsas de galletas a empacar.
- a : un arreglo de longitud k . Para $0 \leq i \leq k - 1$, $a[i]$ indica el número de galletas de tipo i en la alacena.
- La función debe regresar el número de valores distintos de y , tal que la tía Khong pueda empacar x bolsas de galletas con un sabor total de y para cada bolsa.
- La función es llamada un total de q veces (ve las secciones de Límites y Subtareas para los valores permitidos de q). Cada una de estas llamadas debe ser tratada como un escenario separado.

Ejemplos

Ejemplo 1

Considera la siguiente llamada:

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

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

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

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

- una bolsa que contenga tres galletas de tipo 0 y
- dos bolsas, cada un con una galleta de tipo 0 y una galleta de tipo 1.

Como hay 5 posibles valores de y , la función debe regresar 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 Khong quiere empacar 2 bolsas y hay 3 tipos de galletas en la alacena:

- 2 galletas de tipo 0, cada una con un valor de sabor 1,
- 1 galleta de tipo 1, cada una con un valor de sabor 2,
- 2 galletas 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 posibles valores de y , la función debe regresar 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 toda $0 \leq i \leq k - 1$)

- Para cada llamada a `count_tastiness`, la suma de valores de sabor de todas las galletas en la alacena no excede 10^{18} .

Subtareas

1. (9 puntos) $q \leq 10$, y para cada llamada a `count_tastiness`, la suma de valores de sabor de todas las galletas en la alacena 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 de retorno correcto de cada llamada a `count_tastiness` no excede 200 000.
5. (23 puntos) Sin límites adicionales.

Evalrador de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato. La primera línea contiene un entero q . Después, siguen q pares de líneas, 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 evaluador de ejemplo está en el siguiente formato:

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