



Embolsando galletitas (biscuits)

La tía Khong está organizando una competencia con x participantes, y quiere darle una **bolsa de galletitas** a cada uno de ellos. Hay k tipos diferentes de galletitas, numeradas de 0 a $k - 1$. Cada galletita de tipo i ($0 \leq i \leq k - 1$) tiene un **valor de sabrosura** de 2^i . La tía Khong tiene $a[i]$ (puede ser cero) galletitas de tipo i en su despensa.

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

Ayudá a la tía Khong a saber cuántos valores diferentes de y existen tales que es posible llenar x bolsas de galletitas, donde cada una de ellas tiene una sabrosura total de y .

Detalles de implementación

Debes implementar la siguiente función:

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

- x : el número de bolsas de galletitas a llenar.
- a : un arreglo de longitud k . Para todo i tal que $0 \leq i \leq k - 1$, $a[i]$ denota el número de galletitas de tipo i en la despensa.
- La función debe devolver el número de valores diferentes de y , tal que la tía puede llenar x bolsas de galletitas, cada una de ellas teniendo una sabrosura total de y .
- La función es llamada un total de q veces (ver las secciones Cotas y Subtareas para los valores de q permitidos). Cada una de estas llamadas debe ser tratada como un escenario separado.

Ejemplos

Ejemplo 1

Considerá la siguiente llamada:

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

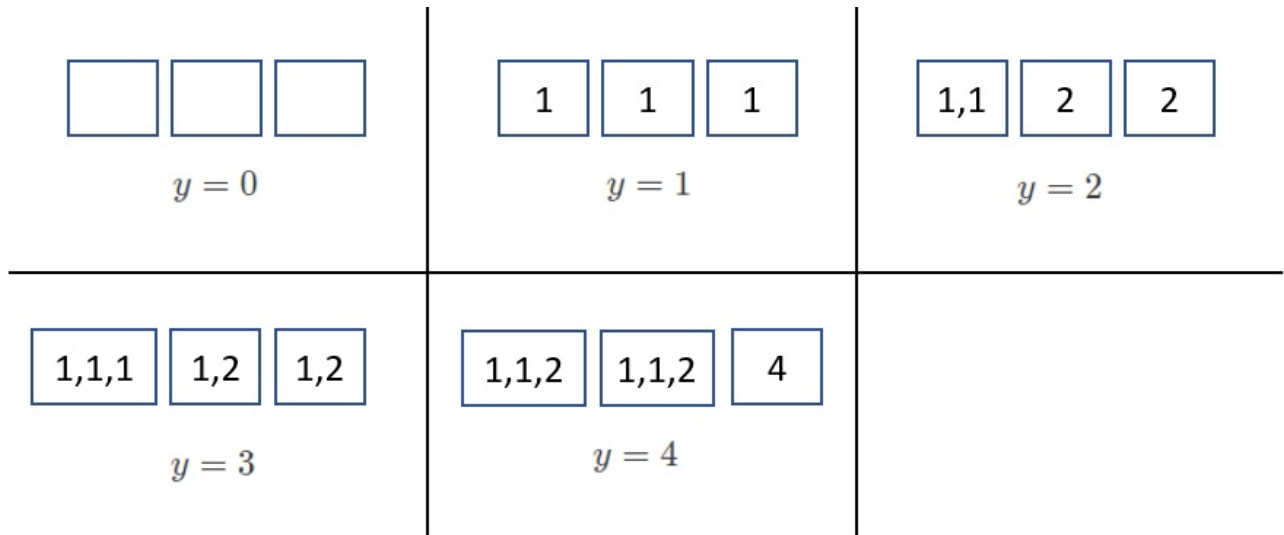
Esto significa que la tía quiere llenar 3 bolsas, y hay 3 tipos de galletitas posibles:

- 5 galletitas de tipo 0, cada una de ellas teniendo un valor de sabrosura 1,
- 2 galletitas de tipo 1, cada una de ellas teniendo un valor de sabrosura 2,
- 1 galletita de tipo 2, cada una de ellas teniendo un valor de sabrosura 4.

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

- una bolsa conteniendo tres galletitas de tipo 0, y
- dos bolsas, cada una conteniendo una galletita de tipo 0 y una galletita de tipo 1.

Como hay 5 valores posibles de y , la función debe devolver 5.



Ejemplo 2

Considera la siguiente llamada:

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

Esto significa que la tía quiere llenar 2 bolsas, y que hay 3 tipos de galletitas posibles:

- 2 galletitas de tipo 0, cada una teniendo un valor de sabrosura 1,
- 1 galletita de tipo 1, cada una teniendo un valor de sabrosura 2,
- 2 galletitas de tipo 2, cada una teniendo un valor de sabrosura 4.

Los valores posibles de y son $[0, 1, 2, 4, 5, 6]$. Como hay 6 valores posibles de y , la función debe devolver 6.

Cotas

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

- Para cada llamada a `count_tastiness` la suma de los valores de sabrosura de todas las galletitas en la despensa no supera 10^{18} .

Subtareas

1. (9 puntos) $q \leq 10$, y para cada llamada a `count_tastiness`, la suma de los valores de sabrosura de todas las galletitas en la despensa no supera 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 respuesta de cada llamada a `count_tastiness` no supera 200 000.
5. (23 puntos) Sin restricciones adicionales.

Evaluador local

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

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

La salida del evaluador local tiene el siguiente formato:

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