Paquets de biscuits (biscuits)

Tante Khong organise une compétition avec x participants, et souhaite donner un **paquet de biscuits** à chaque participant. Il y a k types de biscuits différents, numérotés de 0 à $k-1$. Chaque biscuit de type i ($0 \leq i \leq k-1$) a une **valeur de goût** de 2^i. Tante Khong a $a[i]$ (qui peut valoir zéro) biscuits de type i dans son garde-manger.

Chacun des paquets de Tante Khong contiendra zéro ou plus biscuits de chaque type. Le nombre total de biscuits de type i parmi tous les paquets ne doit pas dépasser $a[i]$. La somme des valeurs de goût de tous les biscuits du paquet est appelé le **goût total** du paquet.

Aidez Tante Khong à trouver combien de valeurs différentes de y existent, telles qu'il est possible de créer x paquets de biscuits, chacun ayant un goût total égal à y.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

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

- x : le nombre de paquets de biscuits à créer.
- a : un tableau de longueur k. Pour $0 \leq i \leq k-1$, $a[i]$ indique le nombre de biscuits de type i dans le garde-manger.
- La fonction doit renvoyer le nombre de valeurs différentes de y, telles que tante Khong peut créer x paquets de biscuits, où chacun a un goût total de y.
- La fonction est appelée un total de q fois (voir les sections Contraintes et Sous-tâches pour les valeurs possibles de q). Chacun de ces appels doit être considéré comme un scénario différent.

Exemples

Exemple 1

Considérons l'appel suivant :

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

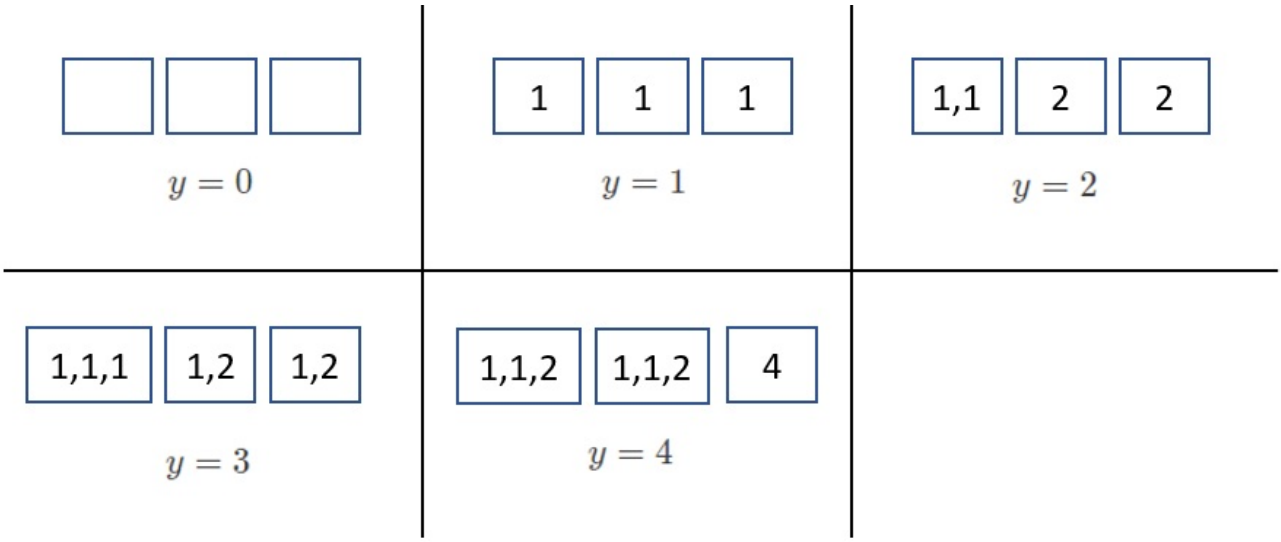
Cela signifie que tante Khong souhaite créer 3 paquets, et qu'il y a 3 types de biscuits dans le garde-manger :

- 5 biscuits de type 0, chacun ayant une valeur de goût de 1,
- 2 biscuits de type 1, chacun ayant une valeur de goût de 2,
- 1 biscuit de type 2, chacun ayant une valeur de goût de 4.

Les valeurs possibles de y sont $[0, 1, 2, 3, 4]$. Par exemple pour créer 3 paquets ayant un goût total de 3, tante Khong peut créer :

- un paquet contenant trois biscuits de type 0, et
- deux paquets, chacun contenant un biscuit de type 0 et un biscuit de type 1.

Comme il y a 5 valeurs possibles pour y, la fonction doit renvoyer 5.



Exemple 2

Considérons l'appel suivant :

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

Cela signifie que tante Khong veut créer 2 paquets, et qu'il y a 3 types de biscuits dans le garde-manger :

- 2 biscuits de type 0, chacun ayant une valeur de goût de 1,
- 1 biscuits de type 1, chacun ayant une valeur de goût de 2,
- 2 biscuits de type 2, chacun ayant une valeur de goût de 4.

Les valeurs possibles pour y sont $[0, 1, 2, 4, 5, 6]$. Comme il y a 6 valeurs possibles pour y, la fonction doit renvoyer 6.

Contraintes

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$

- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$ (pour tout $0 \leq i \leq k-1$)
- Pour chaque appel à `count_tastiness`, la somme des valeurs de goûts de tous les biscuits dans le garde-manger ne dépasse pas 10^{18}.

Sous-tâches

1. (9 points) $q \leq 10$, et pour chaque appel à `count_tastiness`, la somme des valeurs de goût de tous les biscuits dans le garde-manger ne dépasse pas $100\ 000$.
2. (12 points) $x = 1$, $q \leq 10$
3. (21 points) $x \leq 10\ 000$, $q \leq 10$
4. (35 points) La valeur de retour correcte pour chaque appel à `count_tastiness` ne dépasse pas $200\ 000$.
5. (23 points) Pas de contrainte supplémentaire.

Évaluateur d'exemple

L'évaluateur d'exemple lit l'entrée au format suivant. La première ligne contient un entier q. Suivent q paires de lignes, où chaque paire décrit un scénario au format suivant :

- ligne 1 : k x
- ligne 2 : $a[0]$ $a[1]$ $\ldots$ $a[k-1]$

La sortie de l'évaluateur d'exemple est au format suivant :

- ligne i ($1 \leq i \leq q$) : valeur de retour de `count_tastiness` pour le i-ème scenario de l'entrée.