



Packing Biscuits (biscuits)

Tante Khong organiseert een wedstrijd met x deelnemers, en wil elke deelnemer een **tas met koekjes** geven. Er zijn k verschillende soorten koekjes, genummerd van 0 tot en met $k - 1$. Elk koekje van soort i ($0 \leq i \leq k - 1$) heeft een **lekkerheidswaarde** van 2^i . Tante Khong heeft $a[i]$ (mogelijk nul) koekjes van soort i in haar kast.

Elk van Tante Khong's tassen zal nul of meer koekjes van elke soort bevatten. Het totale aantal van koekjes van soort i in alle tassen mag niet groter zijn dan $a[i]$. De som van de lekkerheidswaarde van alle koekjes in een tas heet de **totale lekkerheid** van die tas.

Help Tante Khong om uit te vinden hoe veel verschillende waarden y bestaan, zodat het mogelijk is om x tassen met koekjes in te pakken die ieder dezelfde totale lekkerheid y hebben.

Implementatie details

Je moet de volgende functie implementeren:

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

- x : het aantal tassen met koekjes die moeten worden gevuld.
- a : een array met lengte k . Voor $0 \leq i \leq k - 1$ is $a[i]$ het aantal koekjes van soort i in haar kast.
- De functie moet het aantal verschillende waarden y teruggeven, waarvoor geldt dat Tante x tassen kan inpakken die elk totale lekkerheid y hebben.
- Deze functie wordt q keer aangeroepen (zie randvoorwaarden en subtasks voor de toegestane waarden van q). Elk van deze aanroepen moet worden behandeld als een ander scenario.

Voorbeelden

Voorbeeld 1

Neem de volgende aanroep:

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

Dit betekent dat Tante 3 tassen wil inpakken, en er zijn 3 soorten koekjes in haar kast:

- 5 koekjes van soort 0, elk met een lekkerheidswaarde 1,

- 2 koekjes van soort 1, elk met een lekkerheidswaarde 2,
- 1 koekje van soort 2 met een lekkerheidswaarde 4.

De mogelijke waarden van y zijn $[0, 1, 2, 3, 4]$. Om bijvoorbeeld 3 tassen met elk een totale lekkerheid 3 te maken kan Tante de tassen als volgt inpakken:

- een tas met 3 koekjes van soort 0, en
- twee tassen, elk met één koekje van soort 0 en één koekje van soort 1.

Aangezien er 5 mogelijke waarden zijn voor y moet de functie 5 teruggeven.

<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> $y = 4$	

Voorbeeld 2

Neem de volgende aanroep:

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

Dit betekent dat Tante 3 tassen wil inpakken, en er zijn 3 soorten koekjes in haar kast:

- 2 koekjes van soort 0, elk met een lekkerheidswaarde 1,
- 1 koekje van soort 1 met een lekkerheidswaarde 2,
- 2 koekjes van soort 2, elk met een lekkerheidswaarde 4.

De mogelijke waarden van y zijn $[0, 1, 2, 4, 5, 6]$. Aangezien er 6 mogelijke waarden van y zijn moet de functie 6 teruggeven.

Randvoorwaarden

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

- Voor elke aanroep naar `count_tastiness` zal de totale som van de lekkerheidswaarden van alle koekjes in de kast hoogstens 10^{18} zijn.

Subtasks

1. (9 punten) $q \leq 10$, en voor elke aanroep aan `count_tastiness` zal de som van de lekkerheidswaarden voor alle koekjes in de kast niet boven 10 000 uitkomen.
2. (12 punten) $x = 1, q \leq 10$
3. (21 punten) $x \leq 10\,000, q \leq 10$
4. (35 punten) De juiste teruggegeven waarde van `count_tastiness` zal niet groter zijn dan 200 000.
5. (23 punten) Geen verdere beperkingen.

Voorbeeldgrader

De voorbeeldgrader leest de invoer in het volgende formaat. De eerste regel bevat een integer q . Daarna volgen q paren regels en elk paar omschrijft één scenario in het volgende formaat:

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

De uitvoer van de voorbeeldgrader is in het volgende formaat:

- regel i ($1 \leq i \leq q$): de teruggegeven waarde van `count_tastiness` voor het i -de scenario van de invoer.