Packing Biscuits (biscuits)

Mătușa Khong organizează o competiție cu x participanți și dorește să ofere fiecărui participant câte o **pungă de biscuiți**. Există k tipuri diferite de biscuiți, numerotate de la 0 la $k-1$. Fiecare biscuit de tip i ($0 \leq i \leq k-1$) are o **valoare a aromei** de 2^i. Mătușa Khong are în cămară $a[i]$ (posibil zero) biscuiți de tip i.

Fiecare pungă a Mătușii Khong va conține zero sau mai mulți biscuiți din fiecare tip. Numărul total de biscuiți de tip i din toate pungile nu trebuie să depășească $a[i]$. Suma valorilor aromelor tuturor biscuiților dintr-o pungă se numește **totalul aromelor** pungii.

Ajutați-o pe Mătușa Khong să afle câte valori diferite y există, astfel încât să existe o posibilitate de a împacheta cele x pungi de biscuiți, fiecare pungă având totalul aromelor egal cu y.

Detalii de implementare

Trebuie să implementați următoarea funcție:

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

- x: numărul de pungi de biscuți de împachetat.
- a: un șir de lungime k. Pentru $0 \leq i \leq k-1$, $a[i]$ indică numărul de biscuiți de tip i din cămară.
- Funcția trebuie să returneze numărul de valori distincte y, pentru care Mătușa poate să împacheteze biscuții în x pungi, astfel încât fiecare pungă să aibă totalul aromelor egal cu y.
- Funcția va fi apelată în total de q ori (consultați secțiunile Restricții și Subtaskuri pentru diferitele valori permise ale lui q). Fiecare dintre aceste apeluri trebuie tratat ca un scenariu separat.

Exemple

Exemplul 1

Să considerăm apelul următor:

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

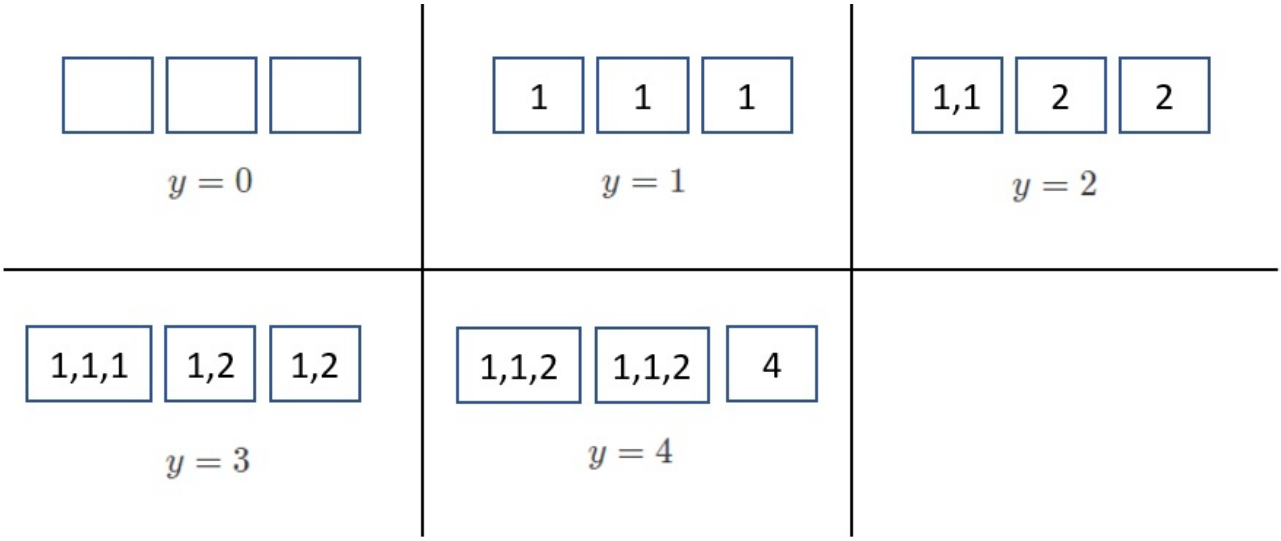
Acesta înseamnă că Mătușa dorește să împacheteze 3 pungi, și există 3 tipuri de biscuiți în magazin:

- 5 biscuiți de tip 0, fiecare având valoarea aromei 1,
- 2 biscuiți de tip 1, fiecare având valoarea aromei 2,
- 1 biscuit de tip 2, fiecare având valoarea aromei 4.

Valorile y posibile sunt $[0, 1, 2, 3, 4]$. De exemplu, pentru a împacheta 3 pungi cu totalul aromelor având valoarea 3, Mătușa poate împacheta:

- o pungă ce conține trei biscuiți de tip 0 și
- două pungi, fiecare conținând câte un biscuit de tip 0 și un biscuit de tip 1.

Deoarece există 5 valori posibile pentru y, funcția trebuie să returneze 5.



Exemplul 2

Să considerăm apelul următor:

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

Acesta înseamnă că Mătușa dorește să împacheteze 2 pungi, și există 3 tipuri de biscuiți în magazin:

- 2 biscuiți de tip 0, fiecare având valoarea aromei 1,
- 1 biscuit de tip 1, fiecare având valoarea aromei 2,
- 2 biscuiți de tip 2, fiecare având valoarea aromei 4.

Valorile y posibile sunt $[0, 1, 2, 4, 5, 6]$. Deoarece există 6 valori posibile pentru y, funcția va trebui să returneze valoarea 6.

Restricții

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$

- $0 \leq a[i] \leq 10^{18}$ (for all $0 \leq i \leq k-1$)
- Pentru fiecare apel al `count_tastiness`, suma valorilor aromelor tuturor biscuiților din cămară nu va depăși 10^{18}.

Subtaskuri

1. (9 puncte) $q \leq 10$, și pentru fiecare apel al `count_tastiness`, valorilor aromelor tuturor biscuiților din cămară nu depășește $100\ 000$.
2. (12 puncte) $x = 1$, $q \leq 10$
3. (21 de puncte) $x \leq 10\ 000$, $q \leq 10$
4. (35 de puncte) Pentru fiecare apel al `count_tastiness` rezultatul corect nu depășește $200\ 000$.
5. (23 de puncte) Fără restricții suplimentare.

Sample grader

Sample graderul citește intrarea în formatul următor. Prima linie conține un întreg q. După aceea, urmează q perechi de linii, fiecare pereche reprezentând câte un scenariu în următorul format:

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

Ieșirea graderului este în următorul format:

- linia i ($1 \leq i \leq q$): valoarea returnată de `count_tastiness` pentru cel de al i-lea scenariu din input.