



Pakovanje Biskvita (biscuits)

Teta Khong organizira takmičenje sa x učenika i želi svakom učeniku pokloniti **kesicu biskvita**. Postoji ukupno k različitih vrsta biskvita, numerisanih od 0 do $k - 1$. Svaki biskvit tipa i ($0 \leq i \leq k - 1$) ima **vrijednost ukusa** od 2^i . Teta Khong u svojoj ostavi ima $a[i]$ (moguće nula) biskvita tipa i .

Svaka kesica tete Khong sadržavaće nula ili više biskvita svake vrste. Ukupan broj biskvita tipa i u svim kesicama ne smije prelaziti $a[i]$. Zbir vrijednosti ukusa svih biskvita u kesici naziva se **ukupan ukus** kesice.

Pomozite teti Khong da sazna koliko različitih vrijednosti y postoji, tako da je moguće spakovati x kesica biskvita, od kojih svaka ima ukupan ukus jednak y .

Detalji Implementacije

Potrebno je implementirati sljedeću proceduru:

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

- x : broj kesica biskvita za pakovanje
- a : niz dužine k . Za svako $0 \leq i \leq k - 1$, $a[i]$ označava broj biskvita tipa i koji se nalaze u ostavi.
- Procedura bi trebala vratiti broj različitih vrijednosti y , tako da teta može spakovati x kesica biskvita, da svaka ima ukupan ukus y .
- Procedura se poziva ukupno q puta (pogledajte Ograničenja i Podzadatke za dopuštene vrijednosti q). Svaki od ovih poziva treba tretirati kao poseban scenarij.

Primjeri

Primjer 1

Posmatrajmo sljedeći poziv:

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

To znači da teta želi spakovati 3 vrećice, i postoje 3 vrste biskvita u ostavi:

- 5 biskvita tipa 0, svaki ima vrijednost ukusa 1,

- 2 biskvita tipa 1, svaki ima vrijednost ukusa 2,
- 1 biskvit tipa 2, koji ima vrijednost ukusa 4.

Moguće vrijednosti y su $[0, 1, 2, 3, 4]$. Na primjer, da bi spakovala 3 vrećice ukupnog ukusa 3, teta može pakovati:

- jedna vrećica koja sadrži 3 biskvita tipa 0, i
- dvije vrećice, gdje svaka sadrži jedan biskvit tipa 0 i jedan biskvit tipa 1.

Budući da postoje 5 mogućih vrijednosti y , procedura treba vratiti 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> $y = 4$	

Primjer 2

Posmatrajmo sljedeći poziv:

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

To znači da teta želi spakovati 2 kesice, i postoje 3 vrste biskvita u ostavi:

- 2 biskvita tipa 0, svaki ima vrijednost ukusa 1,
- 1 biskvit tipa 1, koji ima vrijednost ukusa 2,
- 2 biskvita tipa 2, svaki ima vrijednost ukusa 4.

Potencijalne vrijednosti y su $[0, 1, 2, 4, 5, 6]$. Budući da ima 6 mogućih vrijednosti y , procedura treba vratiti 6.

Ograničenja

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

- Za svaki poziv funkcije `count_tastiness`, suma vrijednosti ukusa svih biskvita u ostavi ne prelazi 10^{18} .

Podzadaci

1. (9 bodova) $q \leq 10$, i za svaki poziv funkcije `count_tastiness`, suma vrijednosti ukusa svih biskvita u ostavi ne prelazi 100 000.
2. (12 bodova) $x = 1, q \leq 10$
3. (21 bod) $x \leq 10\,000, q \leq 10$
4. (35 bodova) Tačne povratne vrijednosti svakog poziva funkcije `count_tastiness` neće prelaziti 200 000.
5. (23 boda) Nema dodatnih ograničenja.

Sample grader

Sample grader čita ulaz u sljedećem formatu. Prva linija sadrži cijeli broj q . Poslije toga slijedi q linija, pri čemu svaka opisuje pojedinačni scenarij u sljedećem formatu:

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

Izlaz sample gradera je u sljedećem formatu:

- linija i ($1 \leq i \leq q$): povratna vrijednost funkcije `count_tastiness` za i -ti scenarij u ulazu.