Membungkus Biskuit (biscuits)

Bibi Khong sedang menjalankan sebuah perlombaan dengan x peserta, dan ingin memberikan setiap peserta **sekantong biskuit**. Terdapat k tipe biskuit berbeda, dinomori dari 0 sampai $k-1$. Setiap biskuit tipe ke-i ($0 \le i \le k-1$) memiliki **nilai kenikmatan** sebesar 2^i. Bibi Khong memiliki $a[i]$ (mungkin nol) biskuit dengan tipe ke-i di dapur.

Setiap kantong Bibi Khong terisi dengan nol atau lebih biskuit dari setiap tipe. Jumlah biskuit dari tipe ke-i di semua kantong tidak boleh melebihi $a[i]$. Jumlah dari nilai kenikmatan setiap biskuit dalam sebuah kantong disebut dengan **total kenikmatan** dari kantong tersebut.

Bantu Bibi Khong mencari tahu ada berapa banyak nilai berbeda dari y yang mungkin, sedemikian sehingga dimungkinkan untuk membungkus x kantong biskuit, masing-masing memiliki total kenikmatan sebesar y.

Detail Implementasi

Anda harus mengimplementasikan fungsi berikut:

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

- x: banyaknya kantong biskuit yang ingin dibungkus.
- a: sebuah *array* dengan panjang k. Untuk $0 \le i \le k-1$, $a[i]$ menunjukkan banyaknya biskuit tipe ke-i di dapur.
- Fungsi ini harus mengembalikan banyaknya nilai y berbeda, sedemikian sehingga Bibi dapat membungkus x kantong biskuit, masing-masing memiliki total kenikmatan sebesar y.
- Fungsi ini akan dipanggil sebanyak q kali (perhatikan bagian Batasan dan Subsoal untuk nilai q). Setiap pemanggilan ini akan dieksekusi sebagai skenario berbeda.

Contoh

Contoh 1

Perhatikan pemanggilan fungsi berikut:

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

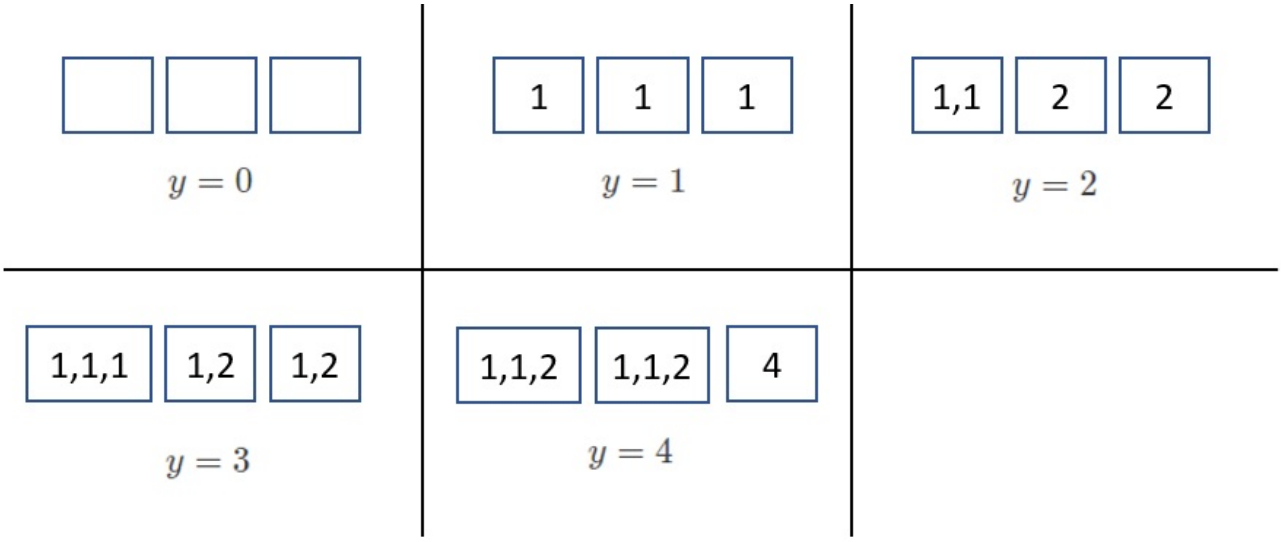
Bibi ingin membungkus 3 kantong, dan terdapat 3 tipe biskuit di dapur:

- 5 biskuit tipe ke-0, masing-masing memiliki nilai kenikmatan sebesar 1,
- 2 biskuit tipe ke-1, masing-masing memiliki nilai kenikmatan sebesar 2,
- 1 biskuit tipe ke-2, masing-masing memiliki nilai kenikmatan sebesar 4.

Nilai y yang mungkin adalah $[0, 1, 2, 3, 4]$. Misalnya, untuk membungkus 3 kantong dengan total kenikmatan 3, Bibi dapat membungkus:

- satu kantong berisikan tiga biskuit tipe ke-0, dan
- dua kantong, masing-masing berisikan satu biskuit tipe ke-0 dan satu biskuit tipe ke-1.

Karena terdapat 5 nilai berbeda yang mungkin untuk y, maka fungsi harus mengembalikan 5.



Contoh 2

Perhatikan pemanggilan fungsi berikut:

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

Bibi ingin membungkus 2 kantong, dan terdapat 3 tipe biskuit di dapur:

- 2 biskuit tipe ke-0, masing-masing memiliki nilai kenikmatan sebesar 1,
- 1 biskuit tipe ke-1, masing-masing memiliki nilai kenikmatan sebesar 2,
- 2 biskuit tipe ke-2, masing-masing memiliki nilai kenikmatan sebesar 4.

Nilai y yang mungkin adalah $[0, 1, 2, 4, 5, 6]$. Karena tedapat 6 nilai berbeda yang mungkin untuk y, fungsi harus mengembalikan 6.

Batasan

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

- Untuk setiap pemanggilan `count_tastiness`, jumlah dari setiap nilai kenikmatan dari semua biskuit di dapur tidak akan melebihi 10^{18}.

Subsoal

1. (9 poin) $q \leq 10$, dan untuk setiap pemanggilan `count_tastiness`, jumlah dari nilai kenikmatan dari semua biskuit di dapur tidak akan melebihi $100\ 000$.
2. (12 poin) $x = 1$, $q \leq 10$
3. (21 poin) $x \leq 10\ 000$, $q \leq 10$
4. (35 poin) Nilai keluaran yang benar dari setiap pemanggilan fungsi `count_tastiness` tidak akan melebihi $200\ 000$.
5. (23 poin) Tidak ada batasan tambahan.

Contoh *grader*

Contoh *grader* akan membaca masukan dengan format sebagai berikut. Baris pertama berisikan sebuah bilangan q. Dilanjutkan dengan q pasang baris berikutnya yang mana setiap pasang baris mendeskripsikan sebuah skenario dengan format sebagai berikut:

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

Keluaran dari contoh *grader* akan dalam format sebagai berikut:

- baris i ($1 \leq i \leq q$): nilai dari pemanggilan `count_tastiness` untuk skenario ke-i dari masukan.