

## بسته‌بندی بیسکویت‌ها (biscuits)

خاله هان، در حال تدارک مسابقه‌ای بین  $x$  شرکت‌کننده است و می‌خواهد به هر شرکت‌کننده، یک بسته‌ی بیسکویت جایزه بدهد.  $k$  نوع مختلف بیسکویت وجود دارد که از 0 تا  $k - 1$  شماره‌گذاری شده‌اند. هر بیسکویت از نوع  $i$  (  $0 \leq i \leq k - 1$  ) یک مقدار خوش‌مزگی برابر با  $2^i$  دارد. خاله هان،  $a[i]$  عدد (شاید صفر) از بیسکویت نوع  $i$  در اختیار دارد.

هر یک از بسته‌های بیسکویت خاله هان، باید صفر یا تعداد بیشتری از هر نوع بیسکویت داشته باشد. مجموع تعداد بیسکویت‌های از نوع  $i$  در تمام بسته‌ها نباید از  $a[i]$  بیشتر شود. مجموع مقدار خوش‌مزگی بیسکویت‌های یک بسته، مجموع خوش‌مزگی بسته نامیده می‌شود.

به خاله هان کمک کنید تا بفهمد، چند مقدار متفاوت  $y$  وجود دارد که می‌توان  $x$  بسته‌ی بیسکویت تهیه کرد که همه مجموع خوش‌مزگی  $y$  داشته باشند.

## جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

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

- $x$ : تعداد بسته‌های بیسکویت که باید تهیه شود.
- $a$ : آرایه‌ای به طول  $k$ . به ازای هر  $0 \leq i \leq k - 1$ ،  $a[i]$  بیانگر تعداد بیسکویت‌های موجود از نوع  $i$  است.
- تابع باید تعداد مقادیر مختلف  $y$ ، که خاله هان می‌تواند  $x$  بسته‌ی بیسکویت هر کدام با مجموع خوش‌مزگی  $y$  تهیه کند را برگرداند.
- این تابع  $q$  مرتبه فراخوانی می‌شود (در قسمت محدودیت‌ها و زیرمسئله‌ها مقادیر مجاز  $q$  را مشاهده کنید). با هر یک از این فراخوانی‌ها، باید به عنوان سناریو مجزا رفتار کرد.

## مثال‌ها

### مثال ۱

فراخوانی زیر را در نظر بگیرید:

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

این یعنی خاله هان می‌خواهد 3 بسته تهیه کند، و 3 نوع بیسکویت در اختیار دارد:

- 5 بیسکویت از نوع 0، هر کدام با مقدار خوش‌مزگی 1،

- 2 بیسکویت از نوع 1، هرکدام با مقدار خوش مزگی 2،
- 1 بیسکویت از نوع 2، هرکدام با مقدار خوش مزگی 4.

مقادیر ممکن  $y$ ،  $[0, 1, 2, 3, 4]$  هستند. به عنوان مثال، برای تهیه 3 بسته با مجموع خوش مزگی 3، خاله هان می‌تواند بسته‌های زیر را تهیه کند:

- یک بسته با سه بیسکویت از نوع 0، و
- دو بسته، هرکدام شامل یک بیسکویت از نوع 0 و یک بیسکویت از نوع 1.

از آن‌جا که 5 مقدار ممکن برای  $y$  وجود دارد، تابع باید 5 را برگرداند.

<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> $y = 4$	

## مثال ۲

فراخوانی زیر را در نظر بگیرید:

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

این یعنی خاله هان می‌خواهد 2 بسته تهیه کند، و 3 نوع بیسکویت در اختیار دارد:

- 2 بیسکویت از نوع 0، هرکدام با مقدار خوش مزگی 1،
- 1 بیسکویت از نوع 1، هرکدام با مقدار خوش مزگی 2،
- 2 بیسکویت از نوع 2، هرکدام با مقدار خوش مزگی 4.

مقادیر ممکن  $y$ ،  $[0, 1, 2, 4, 5, 6]$  هستند. از آن‌جا که 6 مقدار ممکن برای  $y$  وجود دارد، تابع باید 6 را برگرداند.

## محدودیت‌ها

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

- در هر فراخوانی `count_tastiness`، مجموع مقادیر خوش مزگی تمام بیسکویت‌های موجود از  $10^{18}$  بیشتر نیست.

## زیرمسئله‌ها

1. (9 امتیاز)  $q \leq 10$ ، و به ازای هر فراخوانی `count_tastiness`، مجموع مقادیر خوش مزگی تمام بیسکویت‌های موجود از 100 000 بیشتر نیست.
2. (12 امتیاز)  $q \leq 10, x = 1$
3. (21 امتیاز)  $q \leq 10, x \leq 10\,000$
4. (35 امتیاز) پاسخ صحیح هر فراخوانی `count_tastiness` از 200 000 بیشتر نیست.
5. (23 امتیاز) بدون محدودیت اضافه.

## ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند. خط اول شامل یک عدد صحیح به عنوان  $q$  است. سپس،  $q$  جفت از خط‌ها می‌آیند که هر جفت یک سناریو را به شکل زیر توصیف می‌کند:

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

خروجی ارزیاب نمونه در قالب زیر است:

- خط  $i$  ( $1 \leq i \leq q$ ): مقدار بازگردانده شده‌ی `count_tastiness` برای  $i$ -امین سناریوی ورودی. #iscuits