

ビスケットの袋詰め (biscuits)

Aunty Khong は x 人の参加者が参加する大会を開催しており、各参加者に 1 袋ずつ、ビスケットを袋に詰めて渡したい。いま、異なる k 種類のビスケットがあり、 0 から $k - 1$ までの番号が付けられている。種類 i のビスケット ($0 \leq i \leq k - 1$) の美味しさは 2^i である。Aunty Khong の倉庫には、種類 i のビスケットが $a[i]$ 枚 (これは 0 かもしれない) ある。

それぞれの袋には、これらのビスケットのうち何枚かを詰める。1 つの袋には、同じ種類のビスケットが複数枚詰められるかもしれない。すべての袋を合計して、種類 i のビスケットを $a[i]$ 枚を超えて詰めることはできない。1 つの袋に詰められたビスケットの美味しさの総和を **美味しさの和** と言う。

Aunty Khong は美味しさの和をすべての袋について同じにしたい。すべての x 個の袋で美味しさの和をそれぞれ y にできるような y の個数を、Aunty Khong のために求めなさい。

実装の詳細

あなたは以下のプロシーダを実装しなさい。

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

- x : ビスケットを詰める袋の数。
- a : 長さ k の配列である。 $a[i]$ ($0 \leq i \leq k - 1$) は、倉庫にある種類 i のビスケットの枚数である。
- このプロシーダは、すべての x 個の袋で美味しさの和をそれぞれ y にできるような y の個数を戻り値として返さなければならない。
- このプロシーダは、合計 q 回呼び出される。 $(q$ の値については、制約と小課題の節を参照せよ。) それぞれの呼び出しは、別々のシナリオである。

入出力例

入出力例 1

次の呼び出しを考える。

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

これは、Aunty が袋詰めしたい袋の数が 3 つで、倉庫に次の 3 種類のビスケットがあることを表している。

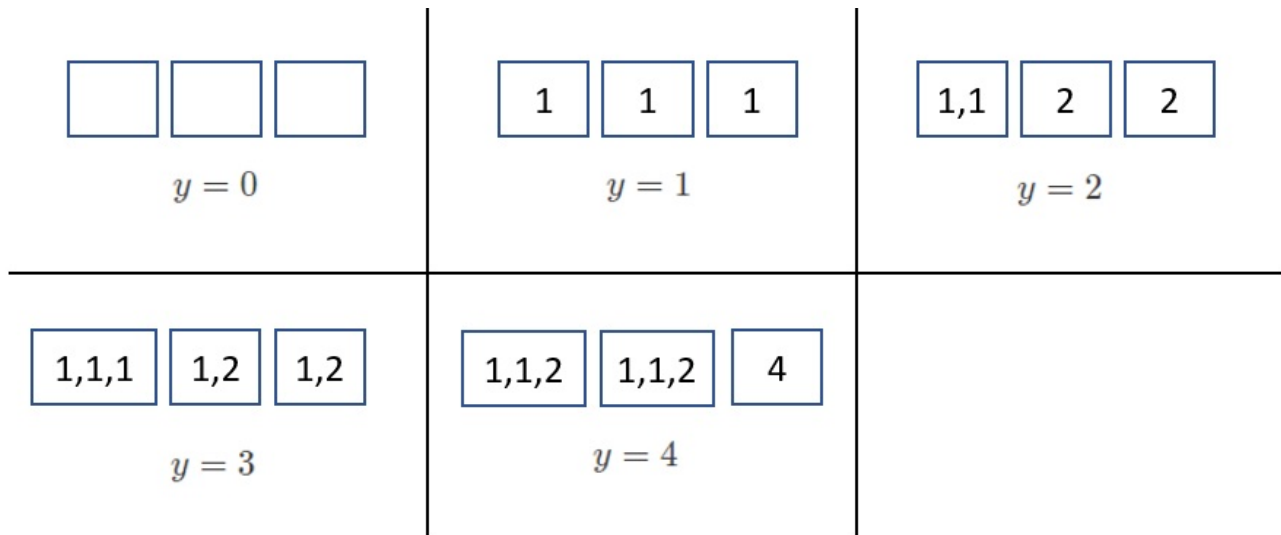
- 種類 0 のビスケットが 5 枚あり、それぞれ美味しさは 1 である。
- 種類 1 のビスケットが 2 枚あり、それぞれ美味しさは 2 である。

- 種類 2 のビスケットが 1 枚あり, それぞれ美味しさは 4 である.

y として可能な値は $[0, 1, 2, 3, 4]$ である. 例えば $y = 3$ を考えると, 美味しさの和がそれぞれ 3 となるような 3 つの袋を作るには, Aunty は次のように袋詰めすれば良い.

- 1 つの袋には, 種類 0 のビスケットを 3 枚詰める.
- 2 つの袋には, 種類 0 のビスケット 1 枚と種類 1 のビスケット 1 枚を詰める.

y として可能な値が 5 個あるので, このプロシージャは 5 を返さなければならない.



入出力例 2

次の呼び出しを考える.

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

これは, Aunty が袋詰めしたい袋の数が 2 つで, 倉庫に次の 3 種類のビスケットがあることを表している.

- 種類 0 のビスケットが 2 枚あり, それぞれ美味しさは 1 である.
- 種類 1 のビスケットが 1 枚あり, それぞれ美味しさは 2 である.
- 種類 2 のビスケットが 2 枚あり, それぞれ美味しさは 4 である.

y として可能な値は $[0, 1, 2, 4, 5, 6]$ である. y として可能な値が 6 個あるので, このプロシージャは 6 を返さなければならない.

制約

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$ ($0 \leq i \leq k - 1$)
- `count_tastiness` のそれぞれの呼び出しについて, 倉庫にあるすべてのビスケットの美味しさの総

和は 10^{18} 以下である.

小課題

1. (9 点) $q \leq 10$. `count_tastiness` のそれぞれの呼び出しについて, 倉庫にあるすべてのビスケットの美味しさの総和は 100 000 以下である.
2. (12 点) $x = 1, q \leq 10$
3. (21 点) $x \leq 10\,000, q \leq 10$
4. (35 点) `count_tastiness` のそれぞれの呼び出しについて, 正しい戻り値は 200 000 以下である.
5. (23 点) 追加の制約は無い.

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を読み込む. 1 行目には整数 q が記されている. その後, 2 行セットの組が q 組続く. 各組は 1 つのシナリオを表しており, 形式は以下の通りである.

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

採点プログラムのサンプルは以下の形式であなたの答えを出力する.

- i 行目 ($1 \leq i \leq q$): 入力の i 番目のシナリオに対応する `count_tastiness` の戻り値.