



裝餅(biscuits)

康阿姨正在組織一個有 x 個參賽者參與的比賽，而每個參賽者都會獲派一袋餅乾。餅乾共有 k 種，編號從 0 到 $k - 1$ 。其中第 i ($0 \leq i \leq k - 1$) 種餅乾的美味值為 2^i 。在康阿姨的茶水間內，第 i 種餅乾她有 $a[i]$ (可能是零) 塊。

康阿姨的每個袋內，每種餅乾都會有零或更多塊。所有袋內的第 i 種餅乾的數量的總和不得超過 $a[i]$ 塊。每袋餅乾中，所有餅乾的美味值總和稱為該袋餅乾的總美味值。

請你幫助康阿姨找出有多少個不同的數值 y 存在，使得她可以把餅乾分成 x 袋，令每袋餅乾的總美味值都恰好等於 y 。

編程實作細節

你應該編寫以下的子程序：

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

- x ：需要分裝餅乾的袋數。
- a ：一個長度為 k 的數組。對於 $0 \leq i \leq k - 1$ ， $a[i]$ 表示在茶水間內第 i 種餅乾有多少塊。
- 子程序應該返回有多少個不同的數值 y ，使得康阿姨可以把餅乾分成 x 袋，令每袋餅乾的總美味值恰好等於 y 。
- 子程序會被總共調用 q 次（請查看限制條件和子任務部份中 q 的數值）。每次調用應當視為一個獨立的情境。

範例

範例1

考慮以下的調用：

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

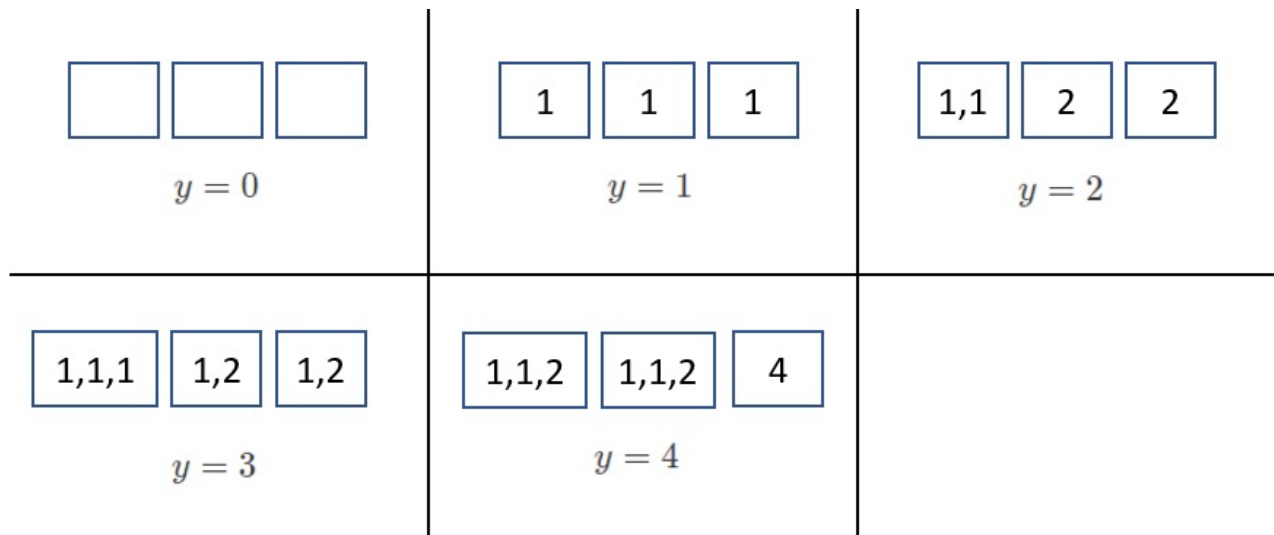
這意味著康阿姨想分裝 3 袋，而在她的茶水間內有 3 種不同的餅乾：

- 第 0 種餅乾有 5 塊，每塊的美味值是 1。
- 第 1 種餅乾有 2 塊，每塊的美味值是 2。
- 第 2 種餅乾有 1 塊，每塊的美味值是 4。

y 的可能數值是 $[0, 1, 2, 3, 4]$ 。例如，想把餅乾分裝成 3 袋，而每袋的總美味值均為 3，則康阿姨可以這樣分裝：

- 第一袋裏有三塊第 0 種的餅乾，和
- 第二袋和第三袋中，每袋有一塊第 0 種餅乾和一塊第 1 種餅乾。

由於 y 有 5 種可能的數值，子程序應該返回 5。



範例 2

考慮以下的調用：

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

這意味著康阿姨想分裝 2 袋，而在她的茶水間內有 3 種不同的餅乾：

- 第 0 種餅乾有 2 塊，每塊的美味值是 1。
- 第 1 種餅乾有 1 塊，每塊的美味值是 2。
- 第 2 種餅乾有 2 塊，每塊的美味值是 4。

y 的可能數值是 $[0, 1, 2, 4, 5, 6]$ 。由於 y 有 6 種可能的數值，子程序應該返回 6。

限制條件

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$ (對於全部 $0 \leq i \leq k - 1$)
- 對於每次調用 `count_tastiness`，茶水間內所有餅乾的美味值相加的總和不會超過 10^{18} 。

子任務

1. (9 分) $q \leq 10$, 且對於每次調用 `count_tastiness`, 茶水間內所有餅乾的美味值的總和不會超過 100 000.
2. (12 分) $x = 1, q \leq 10$
3. (21 分) $x \leq 10\,000, q \leq 10$
4. (35 分) 每次調用 `count_tastiness` 時的正確返回值不會超過 200 000。
5. (23 分) 沒有附加條件。

樣例評分程式

樣例評分程式會讀取以下格式的輸入。第一行會有一個整數 q 。隨後會有 q 組，每組兩行的輸入，表示一個情境：

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

樣例評分程式會用以下格式輸出：

- 第 i ($1 \leq i \leq q$) 行：第 i 個情境中 `count_tastiness` 的返回值。