

Упаковка печенек (biscuits)

Тетя Хонг организует соревнование с x участниками, и хочет дать каждому участнику по **сумке с печеньками**. Есть k разных типов печенек, пронумерованных от 0 до $k - 1$. Каждая из печенек типа i ($0 \leq i \leq k - 1$) имеет **вкусность**, которая равна 2^i . У тети Хонг в кладовке есть $a[i]$ (возможно, 0) печенек типа i .

Каждая сумка тети Хонг должна содержать ноль или более печенек каждого типа. Суммарное количество печенек типа i во всех сумках не должно превосходить $a[i]$. Сумма вкусовностей всех печенек в сумке называется **суммарной вкусовностью** сумки.

Помогите тете Хонг узнать, как много существует различных значений y таких, что существует способ упаковать x сумок с печеньками, у каждой из которых суммарная вкусовность равна y .

Детали реализации

Вам необходимо реализовать следующую функцию:

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

- x : количество сумок с печеньками, которые необходимо упаковать.
- a : массив длины k . Для каждого $0 \leq i \leq k - 1$ значение $a[i]$ обозначает количество печенек типа i в кладовке.
- Функция должна вернуть количество различных значений y таких, что тетя может упаковать x сумок с печеньками, у каждой из которых суммарная вкусовность равна y .
- Функция будет вызвана q раз (разрешенные значения q описаны в секциях Ограничения и Подзадачи). Каждый из этих вызовов должен рассматриваться независимо.

Примеры

Пример 1

Рассмотрим следующий вызов:

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

В этом вызове тетя должна упаковать 3 сумки, и в кладовке есть 3 типа печенек:

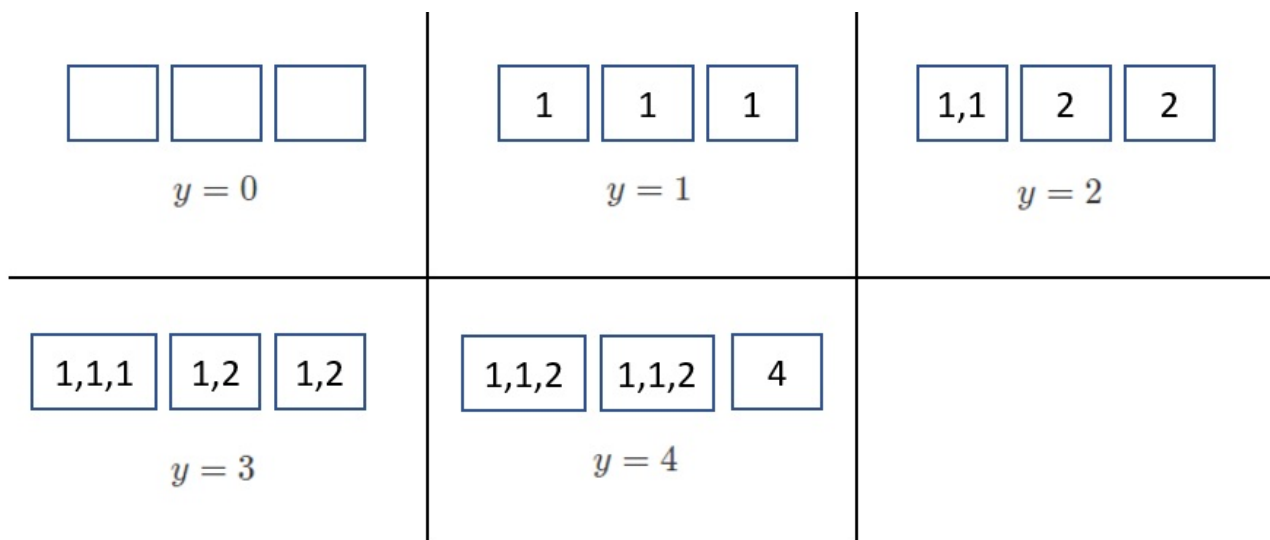
- 5 печенек типа 0, каждая из которых имеет вкусовность 1,

- 2 печеньки типа 1, каждая из которых имеет вкусность 2,
- 1 печеньека типа 2, которая имеет вкусность 4.

Возможные значения y — $[0, 1, 2, 3, 4]$. Например, чтобы упаковать 3 сумки суммарной вкусности 3, тетя может упаковать:

- одну сумку содержащую три печеньки типа 0,
- две сумки, каждая из которых содержит одну печеньку типа 0 и одну печеньку типа 1.

Так как есть 5 возможных значений y , функция должна вернуть 5.



Пример 2

Рассмотрим следующий вызов:

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

В этом вызове тетя хочет упаковать 2 сумки, и в кладовке есть 3 типа печенек:

- 2 печеньки типа 0, каждая из которых имеет вкусность 1,
- 1 печенька типа 1, которая имеет вкусность 2,
- 2 печеньки типа 2, каждая из которых имеет вкусность 4.

Возможные значения y — $[0, 1, 2, 4, 5, 6]$. Так как есть 6 возможных значений y , функция должна вернуть 6.

Ограничения

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

- Для каждого вызова `count_tastiness` сумма вкусовностей всех печенек в кладовке не превосходит 10^{18} .

Подзадачи

1. (9 баллов) $q \leq 10$, и для каждого вызова `count_tastiness` сумма вкусовностей всех печенек в кладовке не превосходит 100 000.
2. (12 баллов) $x = 1, q \leq 10$
3. (21 балл) $x \leq 10\,000, q \leq 10$
4. (35 баллов) Для каждого вызова функции `count_tastiness` значение, которое она должна вернуть, не превосходит 200 000.
5. (23 балла) Нет дополнительных ограничений.

Пример проверяющего модуля

Пример проверяющего модуля считывает данные в следующем формате. Первая строка содержит число q . После этого следует q пар строк, каждая из которых описывает отдельный тестовый случай в следующем формате:

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

Пример проверяющего модуля выводит результат вызовов в следующем формате:

- строка i ($1 \leq i \leq q$): результат функции `count_tastiness` для i -го тестового случая.