

Đổi tiền

Giới hạn thời gian: 1.0s Giới hạn bộ nhớ: 651M

Bạn có n loại tiền xu với mệnh giá lần lượt là $a[1]$, $a[2]$, ..., $a[n]$. Mỗi loại xu được coi là có **vô hạn** số lượng.

Nhiệm vụ của bạn là đếm xem có **bao nhiêu cách khác nhau** để đổi được đúng số tiền S (tổng giá trị các đồng xu bằng S).

Hai cách được coi là **giống nhau** nếu chỉ khác nhau về thứ tự sắp xếp các đồng xu. Nói cách khác, ta chỉ quan tâm đến **số lượng mỗi loại xu**, không quan tâm đến việc lấy xu nào trước, xu nào sau.

Đây là một bài toán kinh điển về **quy hoạch động (Dynamic Programming)**, thường được gọi là bài toán **Coin Change – Counting Solutions**.

Yêu cầu

Cho:

- Số nguyên dương n là số loại tiền xu.
- Dãy $a[1], a[2], \dots, a[n]$ là mệnh giá của từng loại xu.
- Số nguyên dương S là số tiền cần đổi.

Mỗi loại xu có thể dùng **bao nhiêu đồng cũng được** (kể cả 0), miễn sao tổng giá trị đúng bằng S .

Hãy tính **số cách khác nhau** để đổi được đúng số tiền S .

Kết quả có thể rất lớn, tuy nhiên trong bài toán này **không yêu cầu lấy modulo**, hãy in ra **đầy đủ số cách**.

Dữ liệu

- Dòng đầu tiên chứa hai số nguyên dương n, S ($1 \leq n \leq 100, 1 \leq S \leq 10^4$).
- Dòng thứ hai chứa n số nguyên dương $a[1], a[2], \dots, a[n]$ là mệnh giá các đồng xu.

Kết quả

- In ra một số nguyên là **số cách** để đổi được đúng số tiền S bằng các đồng xu đã cho.

Ví dụ

Ví dụ 1

Input

```
3 5
1 2 5
```

Output

```
4
```

Ví dụ 2

Input

```
2 4
1 3
```

Output

```
2
```

Giải thích

Giải thích ví dụ 1

Với $\sim n = 3\sim$, $\sim a = [1, 2, 5]\sim$, $\sim S = 5\sim$, các cách đổi tiền là:

1. Chỉ dùng đồng 5: $\sim\{5\}\sim$
2. Dùng hai đồng 2 và một đồng 1: $\sim\{2, 2, 1\}\sim$
3. Dùng một đồng 2 và ba đồng 1: $\sim\{2, 1, 1, 1\}\sim$
4. Dùng năm đồng 1: $\sim\{1, 1, 1, 1, 1\}\sim$

Tổng cộng có 4 cách, nên kết quả là $\sim 4\sim$.

Giải thích ví dụ 2

Với $n = 2$, $a = [1, 3]$, $S = 4$, ta có các cách đổi tiền sau:

- Dùng bốn đồng 1: 4×1
- Dùng một đồng 1 và một đồng 3: $1 \times 1 + 1 \times 3$

Như vậy tổng cộng có 2 cách, nên kết quả là 2 .

Gợi ý thuật toán Quy hoạch động

Ta có thể định nghĩa:

- $f[x]$ là **số cách** để đổi được đúng số tiền x .

Khởi tạo:

- $f[0] = 1$ (có đúng 1 cách để đạt tổng 0: không chọn đồng xu nào).
- $f[x] = 0$ với mọi $x > 0$ ban đầu.

Sau đó, ta duyệt lần lượt từng loại xu $a[i]$, và cập nhật:

$$\text{for } i = 1 \text{ to } n: \quad \text{for } x = a[i] \text{ to } S: \quad f[x] = f[x] + f[x - a[i]].$$

Ý nghĩa:

- Khi xét đến đồng xu mệnh giá $a[i]$, số cách đạt được x sẽ tăng thêm số cách đạt được số tiền $x - a[i]$ (vì ta thêm một đồng $a[i]$ vào mọi cách cũ tạo ra $x - a[i]$).
- Việc duyệt x tăng dần giúp ta đảm bảo mỗi loại xu được dùng với số lượng bất kỳ nhưng không bị **đếm trùng** các hoán vị.

Cuối cùng, đáp án là $f[S]$.

Độ phức tạp thời gian: khoảng $O(n \cdot S)$, phù hợp với $n \leq 100$, $S \leq 10^4$.

Chăm điểm

Giới hạn:

- $1 \leq n \leq 100$
- $1 \leq S \leq 10^4$
- $1 \leq a[i] \leq S$

Gợi ý phân chia Subtask

Subtask	Giới hạn	Gợi ý thuật toán	Điểm
1	$1 \leq n \leq 20$, $1 \leq S \leq 1000$	Quy hoạch động hai chiều hoặc một chiều	30
2	$1 \leq n \leq 50$, $1 \leq S \leq 5000$	Tối ưu dùng mảng 1 chiều $f[x]$, độ phức tạp $O(n \cdot S)$	30
3	$1 \leq n \leq 100$, $1 \leq S \leq 10^4$	Cài đặt tối ưu, chú ý kiểu dữ liệu (số cách có thể lớn)	40