

Leo cầu thang

Giới hạn thời gian: 1.0s Giới hạn bộ nhớ: 65M

Bạn đang đứng dưới chân một chiếc cầu thang có n bậc. Mỗi lần bước, bạn chỉ được phép:

- Bước lên 1 bậc, hoặc
- Bước lên 2 bậc.

Hỏi có bao nhiêu cách khác nhau để đi từ chân cầu thang lên đến đứng đỉnh (bậc thứ n)?

Đây là một bài toán kinh điển về **quy hoạch động (Dynamic Programming)**, trong đó số cách lên bậc n phụ thuộc vào số cách lên các bậc phía trước nó.

Yêu cầu

Cho một số nguyên dương n là số bậc của cầu thang.
Mỗi lần bạn có thể bước lên 1 bậc hoặc 2 bậc.
Hãy tính **số cách khác nhau** để đi từ chân cầu thang (trước bậc 1) lên đúng bậc thứ n .

Dữ liệu

- Dòng duy nhất chứa một số nguyên n ($1 \leq n \leq 100$) – số bậc của cầu thang.

Kết quả

- In ra một số nguyên là **số cách** để lên đúng bậc thứ n .

Gợi ý kỹ thuật:

- Với n không quá 100 , lời giải sử dụng **quy hoạch động tuyến tính** với độ phức tạp $O(n)$ là hoàn toàn đủ.
- Trong một số ngôn ngữ (như C/C++), kết quả có thể vượt quá giới hạn của kiểu số nguyên 64-bit nếu n lớn. Ngôn ngữ có hỗ trợ số nguyên lớn (như Python) sẽ phù hợp hơn, hoặc đề có thể được điều chỉnh thêm ràng buộc/ghi chú tùy hệ thống chấm.

Ví dụ

Ví dụ 1

Input

2

Output

2

Ví dụ 2

Input

3

Output

3

Giải thích

Giải thích ví dụ 1

Với $n = 2$, ta có các cách đi:

1. Bước 1 bậc, rồi lại 1 bậc: $(1, 1)$
2. Bước 2 bậc một lần: (2)

Tổng cộng có 2 cách, nên kết quả là 2 .

Giải thích ví dụ 2

Với $n = 3$, ta có các cách đi:

1. $(1, 1, 1)$ – bước 1 bậc rồi 1 bậc rồi 1 bậc

- 2. $\sim(1, 2)\sim$ – bước 1 bậc, rồi bước 2 bậc
- 3. $\sim(2, 1)\sim$ – bước 2 bậc, rồi bước 1 bậc

Tổng cộng có 3 cách, nên kết quả là $\sim 3\sim$.

Chấm điểm

Giới hạn:

- $\sim 1 \leq n \leq 100\sim$.

Phân chia Subtask

Subtask	Giới hạn	Gợi ý thuật toán	Điểm
1	$\sim 1 \leq n \leq 20\sim$	Có thể dùng quy hoạch động hoặc đệ quy có nhớ	30
2	$\sim 1 \leq n \leq 50\sim$	Quy hoạch động một chiều, $\sim O(n)\sim$	30
3	$\sim 1 \leq n \leq 100\sim$	Quy hoạch động tối ưu, lưu ý kiểu dữ liệu phù hợp	40