

題目 605

605. 彩帶問題

Submit

Status

Discussion / Solution

Ranklist

Back to Problems List

TopCoder

 Kevin Yu

User's AC Ratio

78.9% (15/19)

Submission's AC Ratio

32.9% (26/79)

Tags

Show solution-related tags

apcscapcsc-recursion


Description





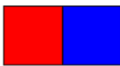

現在 Peipei（裴裴）要請你做出一個漂亮的彩帶，每一段你可以選擇三種顏色，分別是紅，橘，藍三種。至於為何為漂亮呢？漂亮的彩帶有以下的限制：


- 不能有連續的紅色。
- 橘色後面一定要接藍色。




你可以告訴我們如果彩帶的長度為  $n$ ，有幾種可能的漂亮的彩帶嗎？

舉例來說，如果題目輸入 2，你就必須輸出 6，因為答案有以下六種。









官方提示

- 遞迴的參數可以不只一個  $n$ ，再多設一個會好寫很多。

Input Format

輸入只有一行，表示題目的整數  $n$ 。

- $1 \leq n \leq 50$

Output Format

輸出一行一個整數，表示長度  $n$  的漂亮彩帶的可能數。

Sample Input 1

2

Sample Output 1

6

Hints

Problem Source

Subtasks

No.	Testdata Range	Constraints	Score
1	0	範例測資	0
2	0~6	$1 \leq n \leq 22$	30
3	0~11	無額外限制	70

Problem	Total Time (ms)	Max Memory (RSS, KIB) ⓘ	Verdict ⓘ	Score
605. 彩帶問題	48	3456	Accepted	100

Subtask Results				Score
Subtask no.	Testdata Range	Constraints		
1	0	範例測資		0 / 0
2	0~6	$1 \leq n \leq 22$		30 / 30
3	0~11	無額外限制		70 / 70

Testdata Results						
Testdata no.	Subtasks	Time (ms)	Memory (VSS, KIB) ⓘ	Memory (RSS, KIB) ⓘ	Verdict ⓘ	Score
0	1 2 3	3.2	6380	3456	Accepted	100
1	2 3	3.9	6380	3456	Accepted	100
2	2 3	4.1	6380	3456	Accepted	100
3	2 3	4.0	6380	3456	Accepted	100
4	2 3	4.0	6380	3456	Accepted	100
5	2 3	4.1	6380	3456	Accepted	100
6	2 3	4.1	6380	3456	Accepted	100
7	3	4.3	6380	3456	Accepted	100
8	3	4.0	6380	3456	Accepted	100
9	3	4.0	6380	3456	Accepted	100
10	3	3.9	6380	3456	Accepted	100
11	3	3.9	6380	3456	Accepted	100

```
#include <iostream>
using namespace std;

long long dp[1000][3]; // 0:R, 1:O, 2:B

int main() {
    int n;
    cin >> n;

    dp[1][0] = 1; // R
    dp[1][1] = 1; // O
    dp[1][2] = 1; // B

    for (int i = 2; i <= n; i++) {
        // 最後是 R：前面不能是 R
        dp[i][0] = dp[i-1][1] + dp[i-1][2];

        // 最後是 O：不限制，但下一個要是 B，這裡只考慮目前長度
        dp[i][1] = dp[i-1][0] + dp[i-1][2];

        // 最後是 B：可以接任何顏色，也可以是接在 O 後面（處理 O->B）
        dp[i][2] = dp[i-1][0] + dp[i-1][1] + dp[i-2][1];
        // 多加前兩格是因為 O->B 算兩格
    }

    long long ans = dp[n][0] + dp[n][1] + dp[n][2];
    cout << ans << endl;
    return 0;
}
```