

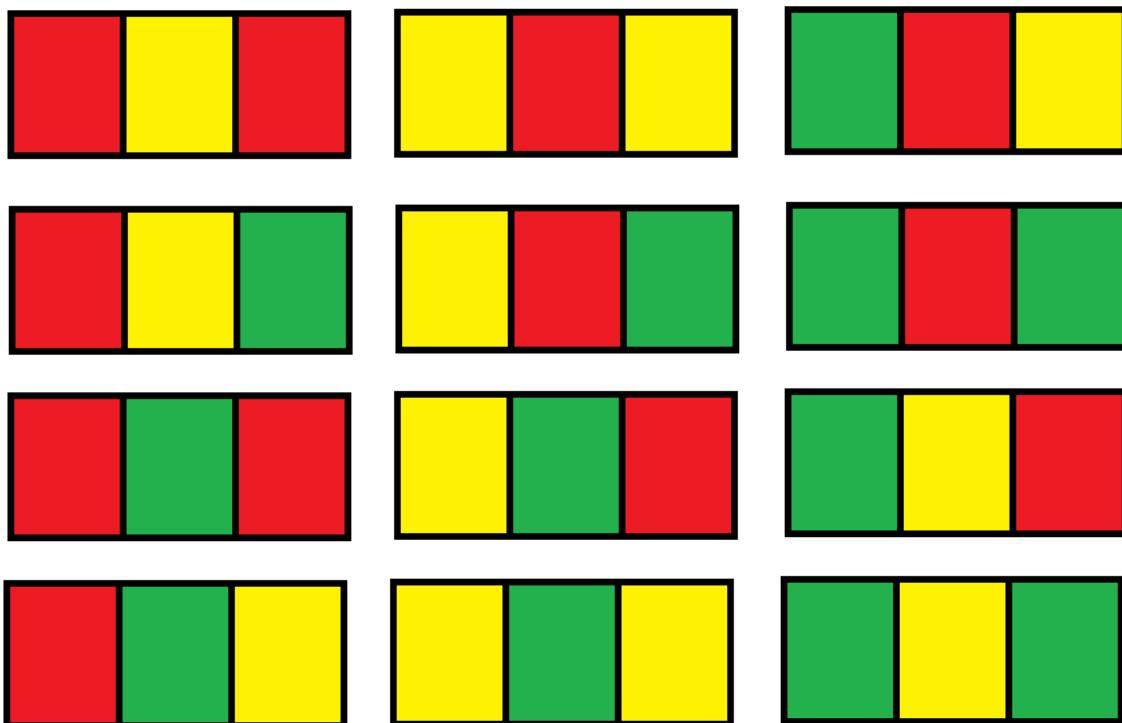
## 1411. Number of Ways to Paint N × 3 Grid

**Statement:-**

You have a grid of size  $n \times 3$  and you want to paint each cell of the grid with exactly one of the three colors: **Red**, **Yellow**, or **Green** while making sure that no two adjacent cells have the same color (i.e., no two cells that share vertical or horizontal sides have the same color).

Given  $n$  the number of rows of the grid, return *the number of ways* you can paint this grid. As the answer may grow large, the answer **must be** computed modulo  $10^9 + 7$ .

**Example 1:**



**Input:**  $n = 1$

**Output:** 12

**Explanation:** There are 12 possible ways to paint the grid as shown.

**Example 2:**

**Input:**  $n = 5000$

**Output:** 30228214

### Constraints:

- $n == \text{grid.length}$
- $1 \leq n \leq 5000$

## Explanation in Hindi

### Problem का मतलब (सरल हिंदी में):

- Grid में **n rows** हैं
- हर row में **3 columns** हैं
- हर cell को तीन रंगों में से एक ही रंग से paint करना है:
  - Red (लाल)
  - Yellow (पीला)
  - Green (हरा)

### Condition (सबसे ज़रूरी नियम):

👉 कोई भी दो adjacent (पास-पास) cells का रंग same नहीं होना चाहिए

### Adjacent का मतलब:

- Horizontal (बाएँ-दाएँ)
- Vertical (ऊपर-नीचे)

Diagonal (तिरछा) से कोई दिक्कत नहीं है।

---

## आपको क्या करना है?

- दिए गए  $n$  के लिए ये count करना है कि  
कुल कितने valid तरीकों से पूरी grid को रंगा जा सकता है
- Answer बहुत बड़ा हो सकता है, इसलिए:

$$\text{Answer} \bmod (10^9 + 7)$$

---

### Example 1:

#### Input:

$n = 1$

#### Explanation:

- सिर्फ 1 row है और 3 columns हैं
- एक row में 3 cells हैं
- तीनों adjacent हैं, इसलिए तीनों का रंग अलग होना चाहिए
- 3 रंगों को 3 जगह arrange करने के कुल 12 valid तरीके बनते हैं

#### Output:

12

---

### Example 2:

#### Input:

$n = 5000$

#### Output:

30228214

यह output program से calculate होता है।

---

## **Summary (short में):**

- Grid:  $n \times 3$
- Colors: 3
- Rule: adjacent cells का color same नहीं
- Task: total valid colorings count करना

## **Essential Points (Short Notes)**

1. **Grid size fixed हैः 3 columns**
  - केवल rows बदलती हैं ( $n$ )
2. **हर row के possible valid patterns पहले से तय होते हैं**
  - एक row में adjacent color same नहीं हो सकता
3. **एक row के total valid patterns = 12**
  - 3 colors, 3 columns, no horizontal clash
4. **दो types के patterns बनते हैं**
  - **Type-A (ABC)** → तीनों color अलग
  - **Type-B (ABA)** → पहला और तीसरा same, बीच वाला अलग
5. **Type-A count = 6**
  - (RGB, RGY, YRG, ...)
6. **Type-B count = 6**
  - (RGR, YRY, GYG, ...)

---

## **Row-to-Row Transition (सबसे important)**

7. **Vertical rule भी follow करना होगा**

- ऊपर वाली row के same column से color match नहीं होना चाहिए

### 8. DP use करते हैं

- $dpA \rightarrow$  ending with Type-A pattern
- $dpB \rightarrow$  ending with Type-B pattern

### 9. Transition formulas

- $newA = dpA * 2 + dpB * 2$
- $newB = dpA * 2 + dpB * 3$

### 10. Base case ( $n = 1$ )

- $dpA = 6$
- $dpB = 6$

### 11. Final answer

- $(dpA + dpB) \% (10^9 + 7)$

Start

|

v

Read n

|

v

Initialize MOD =  $10^9 + 7$

|

v

Base Case (Row = 1)

-----

$dpA = 6$  (ABC type)

$dpB = 6$  (ABA type)

```

|
v
i = 2 to n (Loop)
|
v
newA = (dpA*2 + dpB*2) % MOD
newB = (dpA*2 + dpB*3) % MOD
|
v
dpA = newA
dpB = newB
|
v
Loop End
|
v
Answer = (dpA + dpB) % MOD
|
v
End

```

## Code in java

```

class Solution {

    public int numOfWays(int n) {
        final int MOD = 1_000_000_007;
        long A = 6, B = 6;

        for (int i = 2; i <= n; i++) {
            long newA = (2 * A + 2 * B) % MOD;
            long newB = (2 * A + 3 * B) % MOD;
        }
    }
}

```

```
A = newA;  
B = newB;  
}  
  
return (int) ((A + B) % MOD);  
}  
}
```