**Problem: Painting the Fence**

**Problem Statement**

You are given a fence with n posts and k colors. You have to paint all the posts such that no more than two adjacent fence posts have the same color. Find the total number of ways to paint the fence.

**Scenario**

A landscaper is tasked with painting a long fence in a garden. The fence has multiple posts, and the landscaper has several colors to choose from. However, the garden regulations state that no more than two adjacent posts can have the same color. The landscaper needs to determine the total number of ways to paint the fence while adhering to the rules.

**Input Format**

* The first line contains an integer n, representing the number of posts.
* The second line contains an integer k, representing the number of colors available.

**Constraints**

* 1 <= n <= 50
* 1 <= k <= 50

**Output Format**

* Print the total number of ways to paint the fence.

**Sample Input**

3

2

**Sample Output**

6

**Explanation**

For n = 3 posts and k = 2 colors, the valid ways to paint the fence are:

* Color combinations for 3 posts can be:
  + 1 2 1
  + 1 2 2
  + 2 1 1
  + 2 1 2
  + 1 1 2
  + 2 2 1 Thus, there are 6 ways to paint the fence.

**Additional Test Cases**

**Test Case 1**

4

3

**Output:**

66

**Test Case 2**

2

2

**Output:**

4

**Test Case 3**

5

4

**Output:**

864

**Test Case 4**

1

1

**Output:**

1

**Test Case 5**

10

3

**Output:**

27408

**Solution**

To solve this problem, we can use dynamic programming. We maintain two arrays same and diff:

* same[i]: the number of ways to paint the i-th post the same color as the (i-1)-th post.
* diff[i]: the number of ways to paint the i-th post a different color from the (i-1)-th post.

The recurrence relations are:

* same[i] = diff[i-1]
* diff[i] = (same[i-1] + diff[i-1]) \* (k-1)

The base cases are:

* For i = 1: same[1] = 0 and diff[1] = k

The total number of ways to paint the fence is same[n] + diff[n].

Here's the solution in Python:

python

def countWays(n, k):

if n == 0:

return 0

if n == 1:

return k

same = [0] \* (n + 1)

diff = [0] \* (n + 1)

same[1] = 0

diff[1] = k

for i in range(2, n + 1):

same[i] = diff[i - 1]

diff[i] = (same[i - 1] + diff[i - 1]) \* (k - 1)

return same[n] + diff[n]

# Input

n = int(input().strip())

k = int(input().strip())

# Output

print(countWays(n, k))