

136. Given an  $m \times n$  grid and a ball at a starting cell, find the number of ways to move the ball out of the grid boundary in exactly  $N$  steps.

Example:

Input:  $m=2, n=2, N=2, i=0, j=0$       Output: 6  
Input:  $m=1, n=3, N=3, i=0, j=1$       Output: 12

AIM: To find the path

PROGRAM:

```
def findPaths(m, n, N, i, j):  
  
    MOD = 10**9 + 7  
  
    dp = [[0] * n for _ in range(m)]  
  
    dp[i][j] = 1  
  
    directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]  
  
    count = 0  
  
    for _ in range(N):  
  
        temp = [[0] * n for _ in range(m)]  
  
        for r in range(m):  
  
            for c in range(n):  
  
                for dr, dc in directions:  
  
                    nr, nc = r + dr, c + dc  
  
                    if 0 <= nr < m and 0 <= nc < n:  
  
                        temp[nr][nc] = (temp[nr][nc] + dp[r][c]) % MOD  
  
                    else:  
  
                        count = (count + dp[r][c]) % MOD  
  
        dp = temp  
  
    return count  
  
print(findPaths(2, 2, 2, 0, 0))  
print(findPaths(1, 3, 3, 0, 1))
```

6  
12

OUTPUT :

TIME COMPLEXITY:  $O(m \cdot n \cdot N)$

