EX-1.11

Title :

Count the number of ways to move a ball out of grid boundary in exactly N steps.

Aim:

To design and implement a Python program to find the number of ways to move a ball out of a grid boundary in exactly N steps starting from a given cell.

Procedure:

1. Read the grid dimensions `m` and `n`, number of steps `N`, and starting cell coordinates `(i, j)`.
2. Use dynamic programming with memoization to efficiently compute the number of ways:
   - Define a recursive function `dfs(x, y, steps_left)` that returns the number of ways to move out of boundary starting at `(x, y)` with `steps_left` moves remaining.
   - If the ball is out of boundary, return 1 indicating a valid path.
   - If no steps remain and ball is inside, return 0.
   - Otherwise, recursively try moving up, down, left, and right, summing the paths, and memoize results to avoid recomputation.
3. Use modulo 109+7 to keep numbers within limit.
4. Print the total number of ways.

Algorithm:

1. Start
2. Input `m, n, N, i, j`
3. Create a memo dictionary to store results for `(x, y, steps_left)` states.
4. Define recursive DFS function:
   - If `(x, y)` is out of boundary, return 1.
   - If `steps_left == 0`, return 0.
   - If result in memo, return it.
   - Compute sum of DFS calls for (x+1, y), (x-1, y), (x, y+1), (x, y-1) with `steps_left-1`.
   - Store result modulo 109+7 in memo.
   - Return result.
5. Call DFS with `(i, j, N)`.
6. Print result.
7. Stop

Input:

2 2 2 0 0

1 3 3 0 1

Output:

6

12

Program :

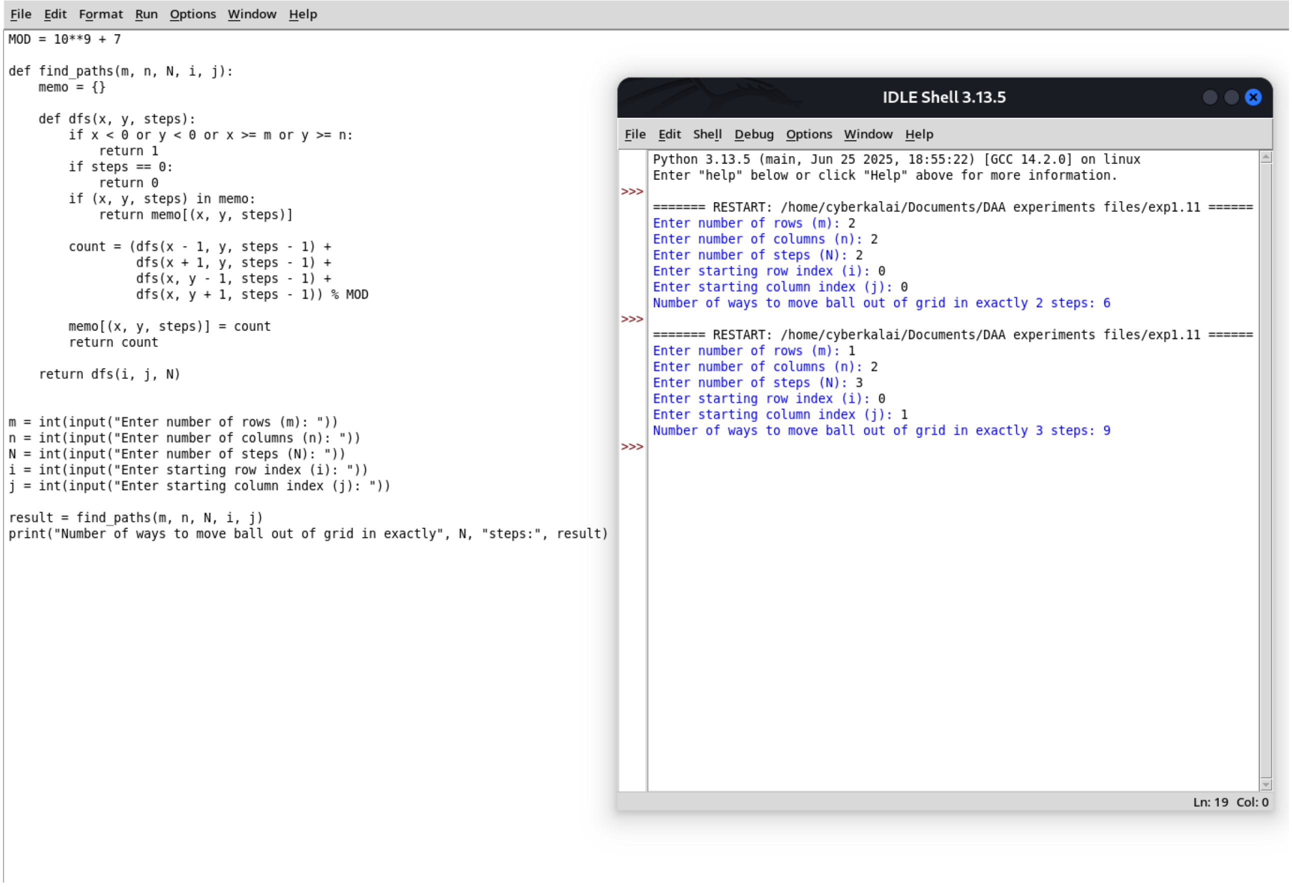
```
MOD = 10**9 + 7
def findPaths(m, n, maxMove, startRow, startColumn):
    memo = {}
    def dfs(x, y, moves_left):
        if x < 0 or x >= m or y < 0 or y >= n:
            return 1
        if moves_left == 0:
            return 0
        if (x, y, moves_left) in memo:
            return memo[(x, y, moves_left)]
        paths = (dfs(x + 1, y, moves_left - 1) +
                 dfs(x - 1, y, moves_left - 1) +
                 dfs(x, y + 1, moves_left - 1) +
                 dfs(x, y - 1, moves_left - 1)) % MOD
        memo[(x, y, moves_left)] = paths
        return paths
    return dfs(startRow, startColumn, maxMove)
m, n, N, i, j = map(int, input("Enter m n N i j: ").split())
result = findPaths(m, n, N, i, j)
print("Number of ways:", result)
```

Performance Analysis:

Time Complexity: O(m * n * N)

Space Complexity: O(m* n * N)

program output:



File Edit Format Run Options Window Help
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Ln: 19 Col: 0

```
MOD = 10**9 + 7

def find_paths(m, n, N, i, j):
    memo = {}

    def dfs(x, y, steps):
        if x < 0 or y < 0 or x >= m or y >= n:
            return 1
        if steps == 0:
            return 0
        if (x, y, steps) in memo:
            return memo[(x, y, steps)]

        count = (dfs(x - 1, y, steps - 1) +
                 dfs(x + 1, y, steps - 1) +
                 dfs(x, y - 1, steps - 1) +
                 dfs(x, y + 1, steps - 1)) % MOD

        memo[(x, y, steps)] = count
        return count

    return dfs(i, j, N)


m = int(input("Enter number of rows (m): "))
n = int(input("Enter number of columns (n): "))
N = int(input("Enter number of steps (N): "))
i = int(input("Enter starting row index (i): "))
j = int(input("Enter starting column index (j): "))

result = find_paths(m, n, N, i, j)
print("Number of ways to move ball out of grid in exactly", N, "steps:", result)
```

```
Python 3.13.5 (main, Jun 25 2025, 18:55:22) [GCC 14.2.0] on linux
Enter "help" below or click "Help" above for more information.
>>>
======= RESTART: /home/cyberkalai/Documents/DAA experiments files/exp1.11 ======
Enter number of rows (m): 2
Enter number of columns (n): 2
Enter number of steps (N): 2
Enter starting row index (i): 0
Enter starting column index (j): 0
Number of ways to move ball out of grid in exactly 2 steps: 6
>>>
======= RESTART: /home/cyberkalai/Documents/DAA experiments files/exp1.11 ======
Enter number of rows (m): 1
Enter number of columns (n): 2
Enter number of steps (N): 3
Enter starting row index (i): 0
Enter starting column index (j): 1
Number of ways to move ball out of grid in exactly 3 steps: 9
>>>
```

Result :

Thus the given program Out of Boundary Paths is executed and got output successfully.