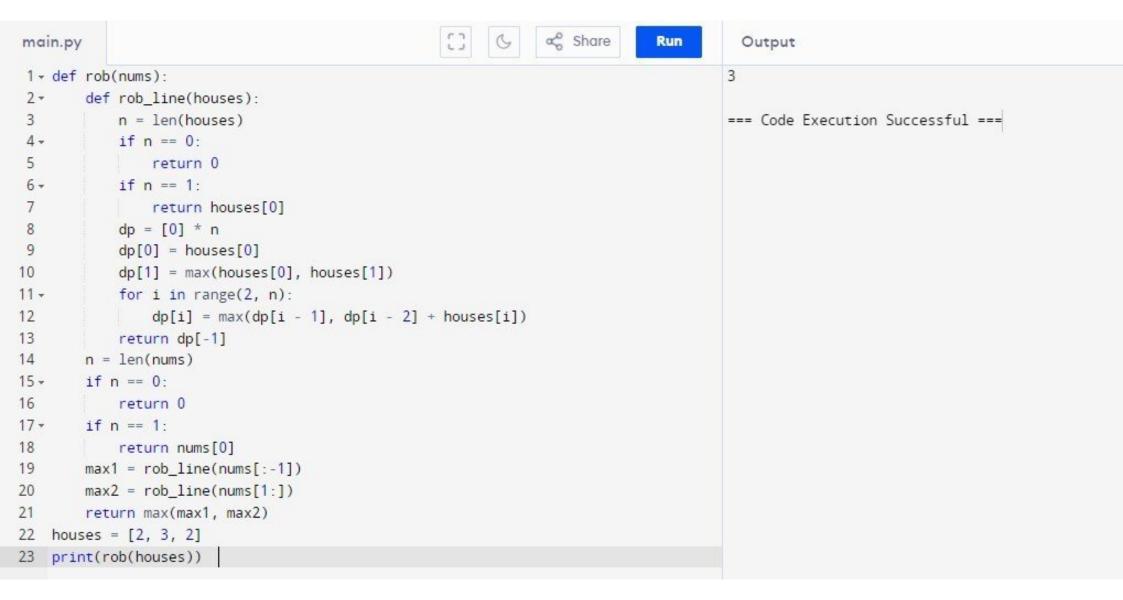
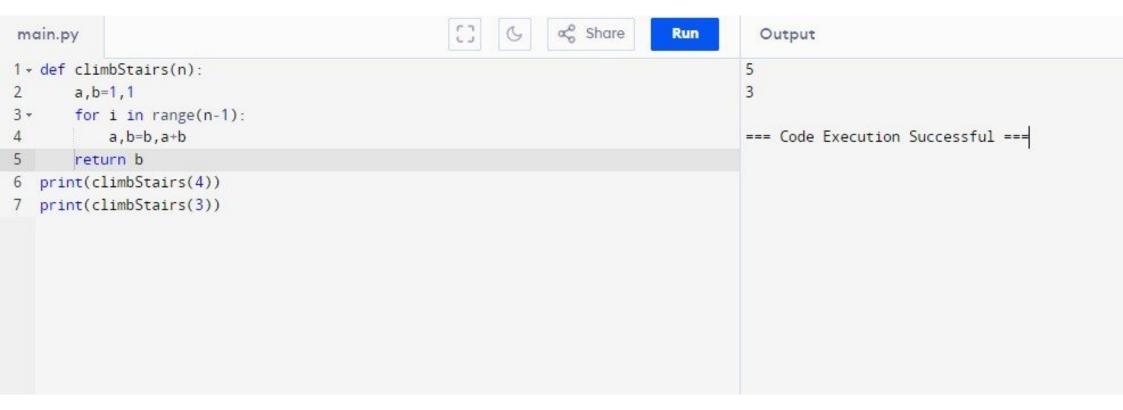
```
∝ Share
main.py
                                                                    Run
                                                                              Output
 1 - def findWays(m, n, N, i, j):
                                                                            6
        dp = [[[0 for _ in range(N+1)] for _ in range(n)] for _ in range(m
 2
                                                                            === Code Execution Successful ===
            )]
        for step in range(1, N+1):
 3 -
            for x in range(m):
 4 -
                for y in range(n):
                    ways = (
                        (dp[x-1][y][step-1] if x > 0 else 1) +
                        (dp[x+1][y][step-1] if x < m-1 else 1) +
                        (dp[x][y-1][step-1] if y > 0 else 1) +
10
                        (dp[x][y+1][step-1] if y < n-1 else 1)
11
12
                    dp[x][y][step] = ways
        return dp[i][j][N]
13
14 m = 2
   n = 2
   N = 2
17 i = 0
18 j = 0
19 print(findWays(m, n, N, i, j))
```









```
main.py
                                                             C 3 € ac Share
                                                                                                 Output
1 - def gameOfLife(board):
                                                                                                [[0, 0, 0], [1, 0, 1], [0, 1, 1], [0, 1, 0]]
       m, n = len(board), len(board[0])
        directions = [(-1, -1), (-1, 0), (-1, 1),
                                                                                                === Code Execution Successful ===
                    (0, -1), (0, 1),
                   (1, -1), (1, 0), (1, 1)
       def countLiveNeighbors(r, c):
           count = 0
           for dr, dc in directions:
               nr, nc = r + dr, c + dc
 9
               if 0 \le nr \le m and 0 \le nc \le n and abs(board[nr][nc]) == 1:
10 -
                   count += 1
11
12
           return count
       for r in range(m):
13 -
14 -
            for c in range(n):
               live_neighbors = countLiveNeighbors(r, c)
15
               if board[r][c] == 1 and (live_neighbors < 2 or live_neighbors > 3):
16 -
                   board[r][c] = -1 # Live to dead
17
               if board[r][c] == 0 and live_neighbors == 3:
18 -
19
                   board[r][c] = 2 # Dead to live
       for r in range(m):
20 -
           for c in range(n):
21 -
               if board[r][c] == -1:
22 -
                   board[r][c] = 0
23
               if board[r][c] == 2:
24 -
25
                   board[r][c] = 1
26 board = [ [0, 1, 0], [0, 0, 1], [1, 1, 1], [0, 0, 0]]
27 gameOfLife(board)
28 print(board)
```

