

```

class Solution:
    def spiralOrder(self, matrix: List[List[int]]) -> List[int]:
        if not matrix:
            return []
        m, n = len(matrix), len(matrix[0])
        seen = [[False] * n for _ in matrix]
        ans = []
        dr, dc = [0, 1, 0, -1], [1, 0, -1, 0]
        r = c = di = 0
        for _ in range(m * n):
            ans.append(matrix[r][c])
            seen[r][c] = True
            cr, cc = r + dr[di], c + dc[di]
            if 0 <= cr < m and 0 <= cc < n and not seen[cr][cc]:
                r, c = cr, cc
            else:
                di = (di + 1) % 4
                r, c = r + dr[di], c + dc[di]
        return ans

```

```

class Solution:
    def subarraySum(self, nums: List[int], k: int) -> int:
        count = 0
        prefix_sum = {0: 1}
        sum_so_far = 0
        for num in nums:
            sum_so_far += num
            if sum_so_far - k in prefix_sum:
                count += prefix_sum[sum_so_far - k]
            prefix_sum[sum_so_far] = prefix_sum.get(sum_so_far, 0) + 1
        return count

```