



// solution

class Solution:

def shiftGrid(self, grid: List[List[int]], k: int) -> List[List[int]]:

# corner cases

T = len(grid)\*len(grid[0])

if k == 0 or k == T: return grid

# unpack the list

flat = [element for row in grid for element in row]

# rotate the unpacked list

if k > T: k %= T

right, left = flat[T-k:], flat[:T-k]

flat = right + left

# pack the rotated list

i = 0

for r in range(len(grid)):

for c in range(len(grid[0])):

grid[r][c] = flat[i]

i += 1

return grid

faster than 95%