import sys

def Check(a):

if a > 1:

for x in range(2, int(a/2) + 1):

if (a % x) == 0:

return False

break

else:

return True

else:

return False

def rotateMatrix(mat):

if not len(mat):

return 0

top = 0

bottom = len(mat)-1

left = 0

right = len(mat[0])-1

while left < right and top < bottom:

prev = mat[top][right]

if(Check(int(mat[top][left]))==False):

top,bottom,left,right,prev,mat=antiClockwise(top,bottom,left,right,prev,mat)

else:

top,bottom,left,right,prev,mat=Clockwise(top,bottom,left,right,prev,mat)

return mat

def antiClockwise(top,bottom,left,right,prev,mat):

for i in range(right,left-1,-1):

curr = mat[top][i]

mat[top][i] = prev

prev = curr

top +=1

for i in range(top, bottom+1):

curr = mat[i][left]

mat[i][left] = prev

prev = curr

left +=1

for i in range(left, right+1):

curr = mat[bottom][i]

mat[bottom][i] = prev

prev = curr

bottom -= 1

for i in range(bottom, top-2, -1):

curr = mat[i][right]

mat[i][right] = prev

prev = curr

right -=1

return [top,bottom,left,right,prev,mat]

def Clockwise(top,bottom,left,right,prev,mat):

for i in range(left, right+1):

curr = mat[top][i]

mat[top][i] = prev

prev = curr

top += 1

for i in range(top, bottom+1):

curr = mat[i][right]

mat[i][right] = prev

prev = curr

right -= 1

for i in range(right, left-1, -1):

curr = mat[bottom][i]

mat[bottom][i] = prev

prev = curr

bottom -= 1

for i in range(bottom, top-2, -1):

curr = mat[i][left]

mat[i][left] = prev

prev = curr

left += 1

return [top,bottom,left,right,prev,mat]

def printMatrix(mat):

for row in mat:

for col in row:

print (col, end = " ")

print()

matrix = []

rows = int(input())

if((1<=rows<=10\*\*7)==False):

print("INVALID INPUT")

sys.exit()

cols = int(input())

if((1<=cols<=10\*\*7)==False):

print("INVALID INPUT")

sys.exit()

for i in range(0,rows):

addl=list([int(item) for item in input().split()])

for x in addl:

if((1<=x<=10\*\*5)==False):

print("INVALID INPUT")

sys.exit()

matrix.append(addl)

matrix = rotateMatrix(matrix)

printMatrix(matrix)