## LIITE: Mondrian-ruudukkojen koodi

def fill(y, x, h, w, c):

 for i in range(h):

 for j in range(w):

 grid[y+i][x+j] = c

def empty(y, x, h, w):

 for i in range(h):

 for j in range(w):

 if grid[y+i][x+j]:

 return False

 return True

def search(y, x, count, amin, amax):

 global best

 if amax-amin >= best:

 return

 if y == n:

 if count > 1 and amax-amin < best:

 best = amax-amin

 print("Uusi tulos: virhe", best)

 for i in range(n):

 print(grid[i])

 elif x == n:

 search(y+1, 0, count, amin, amax)

 elif grid[y][x]:

 search(y, x+1, count, amin, amax)

 else:

 for h in range(1,n-y+1):

 for w in range(1, n-x+1):

 if not used[h][w] and empty(y, x, h, w):

 used[h][w] = used[w][h] = 1

 fill(y, x, h, w, count+1)

 search(y, x+1, count+1, min(amin, h\*w), max(amax, h\*w))

 used[h][w] = used[w][h] = 0

 fill(y, x, h, w, 0)

 else:

 break

n = int(input("Anna n: "))

best = n\*n

grid = [[0]\*n for \_ in range(n)]

used = [[False]\*(n+1) for \_ in range(n+1)]

search(0, 0, 0, n\*n, 0)

print("Paras tulos: virhe", best)