# 復習済み

A-E

# A aaaadaa

正解

# B ARC Division

正解

# C Perfect Standings

正解

# D Repeated Sequence

正解

配列Aの合計で割ったあまりが、Aが二つ繋がった配列の連続部分列の合計に存在するかをしゃくとり法で確認する。

# E Takahashi is Slime 2

不正解

隣接するスライムを強さのヒープで管理して小さいやつから合成していく。

注意として power\*X >= strengthはpower >= strength/Xにすると桁が小さい部分が判定されずに間違った答えを返し得る。

## 正解コード

from heapq import heapify, heappop, heappush

H, W, X = map(int, input().split())

P, Q = map(int, input().split())

P -= 1

Q -= 1

S = [list(map(int, input().split())) for \_ in range(H)]

check = [[False]\*W for \_ in range(H)]

check[P][Q] = True

strength = S[P][Q]

q = []

heapify(q)

for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:

ni = P+di

nj = Q+dj

if 0<=ni<H and 0<=nj<W and check[ni][nj] == False:

check[ni][nj] = True

heappush(q, (S[ni][nj], ni, nj))

while q:

power, i, j = heappop(q)

if power\*X >= strength:

break

strength += power

for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:

ni = i+di

nj = j+dj

if 0<=ni<H and 0<=nj<W and check[ni][nj] == False:

check[ni][nj] = True

heappush(q, (S[ni][nj], ni, nj))

print(strength)

# F Double Sum 2

解いてない

# G Abs Sum

解いてない