PART\_B

CODE:

target = {

2: 1,

3: 2,

4: 3,

5: 4,

6: 5,

7: 6,

8: 5,

9: 4,

10: 3,

11: 2,

12: 1

}

dieA = [1, 2, 3, 4, 0, 0]

dieB = [1, 3, 4, 5, 6, 0]

def validate\_cur\_combination():

cur = {}

for i in range(6):

for j in range(6):

if dieA[i] + dieB[j] in cur:

cur[dieA[i] + dieB[j]] += 1

else:

cur[dieA[i] + dieB[j]] = 1

return cur == target

def print\_die():

print("Die A is:", dieA)

print("Die B is:", dieB)

def backtrack(curA, curB):

if curA == 6 and curB == 6:

if validate\_cur\_combination():

print\_die()

return True

return False

for k in range(4):

curAHit = k + 1

for l in range(11):

curBHit = l + 1

dieA[curA] = curAHit

dieB[curB] = curBHit

if backtrack(curA + 1, curB + 1):

return True

return False

if \_\_name\_\_ == "\_\_main\_\_":

if not backtrack(4, 4):

print("No valid combination found.")