#### The Doomed Dice Challenge

Logic to the problem

#### 1. Total Combinations:

The total combinations when rolling two six-sided dice can be calculated by multiplying the number of faces on each die. Since both Die A and Die B have 6 faces each, the total combinations (C) can be expressed as:

*C*=Number of faces on Die A×Number of faces on Die B=6×6=36

#### 2. Distribution of Combinations:

We can create a 6x6 matrix to represent all possible combinations when rolling both Die A and Die B. Each cell

[*i*][*j*] in the matrix represents the combination where Die A shows face *i*+1 and Die B shows face *j*+1.

#### 3. Probability of Sums:

The probability of obtaining a particular sum is the ratio of the number of combinations resulting in that sum to the total number of combinations.

# Calculating the probability of each sum

total\_combinations = 36

for i in range(2, 13):

sum\_count = sum(1 for row in combinations\_matrix for num in row if num == i)

probability = sum\_count / total\_combinations

print(f"P(Sum = {i}) = {probability}")

#### Transform Function undoom\_dice:

Now, we need to create a function that transforms the dice according to the conditions given by Loki.

def undoom\_dice(die\_a, die\_b):

# Transform Die A

new\_die\_a = [min(num, 4) for num in die\_a]

# Transform Die B (no restrictions)

new\_die\_b = die\_b

return new\_die\_a, new\_die\_b

This function takes Die A and Die B as input and outputs the transformed dice, ensuring that Die A has no more than 4 spots on each face.