**Logic of the Code: PART B**

Calculate Probability:

The calculateProbability method computes the probability distribution of sums for a given pair of dice (dieA and dieB).

It uses an integer array (probabilityArray) to store the count of occurrences for each sum.

**Check if All Spots are Valid:** The allSpotsValid method ensures that all spots on a die are valid, i.e., not greater than 4.

**Check if Arrays are Equal:** The arraysEqual method checks whether two integer arrays are equal.

**Undoom Dice Function:** The undoomDice function attempts to find a new configuration for New\_Die\_A that satisfies the conditions imposed by Loki: It iterates through each face of the original Die\_A.

For each face, it tries new configurations of spots (1 to 4, excluding the original configuration).

It checks if the probability distribution remains the same as the original and if all spots are valid.

If a valid configuration is found, it returns the new configuration for New\_Die\_A.

**How the Solution Was Formulated:**

**Understanding the Problem:**

The problem requires finding a new configuration for Die\_A after it has been "doomed" by Loki.

Constraints include that no face on Die\_A can have more than 4 spots, and the probability distribution of sums with Die\_B must remain the same.

**Probability Calculation:**

The initial approach was to calculate the probability distribution for the original dice (Die\_A and Die\_B).

This information is essential for ensuring that the probability distribution remains unchanged after reattaching spots.

**Iterative Approach:**

The solution employs an iterative approach, where each face of the original Die\_A is considered, and various configurations of spots are tested.

The goal is to find a valid configuration that maintains the same probability distribution.

**Array Manipulation:**

Arrays are used for simplicity and efficiency.

The arraysEqual method is introduced to compare two arrays for equality.

**Validation and Testing:**

The solution includes methods to check if spots are valid and if two arrays are equal.

It also includes testing for a solution. If a valid configuration is found, it is returned; otherwise, null is returned.

By breaking down the problem into smaller components and addressing each requirement step by step, the solution was formulated to meet the specified conditions and constraints. Testing with different inputs ensures the correctness and effectiveness of the implemented solution.