**---------------------------------------PART – A--------------------------------------**

**Problem Statement Overview:**

The problem involves analyzing two six-sided dice (Die A and Die B) to calculate total combinations, list all possible combinations, and determine the probability of each sum.

**Question - 1:**

**How many total combinations are possible? Show the math along with the code!**

**Logic**:

The total number of combinations possible when rolling both Die A and Die B together, we multiply the number of faces on Die A with the number of faces of Die B.

So, the total combinations: 6 \* 6 = 36

**Question - 2:**

**Calculate and display the distribution of all possible combinations that can be obtained when rolling both Die A and Die B together. Show the math along with the code!**

**Logic for generating all possible combinations:**

1. To generate all possible combination pairs, we iterate through each face of Die A and Die B.
2. For example, if Die A has values [1, 2] and Die B has values [3, 4, 5], the pairs generated will be (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), and (2, 5).

**Logic for generating the sums:**

1. For each combination of values (a, b) from Die A and Die B, we compute their sum as a + b.
2. We store these sums in a 2D matrix where each cell (i, j) contains the sum of the ith value from Die A and the jth value from Die B.

**Question - 3:**

**Calculate the Probability of all Possible Sums occurring among the number of combinations from (2).**

**Logic:**

* Utilize the ‘getFreq’ method to calculate the frequency of each sum.
* The probability of each sum is calculated by dividing its frequency by the total number of combinations.
* **Math:** P(sum)=frequency/total number of combinations

**----------------------------------------PART - B--------------------------------------------**

**Question:**

**Now comes the real challenge. You were happily spending a lazy afternoon playing**

**your board game with your dice when suddenly the mischievous Norse God Loki ( You**

**love Thor too much & Loki didn’t like that much ) appeared.**

**Loki dooms your dice for his fun removing all the “Spots” off the dice.**

**No problem! You have the tools to re-attach the “Spots” back on the Dice.**

**However, Loki has doomed your dice with the following conditions:**

**● Die A cannot have more than 4 Spots on a face.**

**● Die A may have multiple faces with the same number of spots.**

**● Die B can have as many spots on a face as necessary i.e. even more than 6.**

**But in order to play your game, the probability of obtaining the Sums must remain the**

**same!**

**So if you could only roll P(Sum = 2) = 1/X, the new dice must have the spots reattached**

**such that those probabilities are not changed.**

**Input:**

**● Die\_A = [1, 2, 3, 4, 5, 6] & Die B = Die\_A = [1, 2, 3, 4, 5, 6]**

**Output:**

**● A Transform Function undoom\_dice that takes (Die\_A, Die\_B) as input &**

**outputs New\_Die\_A = [?, ?, ?, ?, ?, ?],New\_Die\_B = [?, ?,**

**?, ?, ?, ?] where,**

**● No New\_Die A[x] > 4**

**LOGIC:**

**1.** **Generating Combinations:**

* Two nested loops are used to iterate over all possible values for each die, generating all combinations of dice values.
* Each combination is stored as a list of integers representing the values of the two dice.

**2. Sorting Combinations:**

* After generating all combinations, they are sorted based on the sum of their values.
* This sorting ensures that combinations with smaller sums appear first in the list, while combinations with larger sums appear later.

**3. Modifying Combinations:**

* Each combination is examined, and if the value of the first die (valueA) is less than or equal to 4, it remains unchanged.
* If valueA is greater than 4, the values of the two dice are swapped. This modification ensures that valueB (previously valueA) becomes the value for the first die, while valueA (previously valueB) becomes the value for the second die.

**4. Output:**

* The modified combinations are stored in a 2D array, where each row represents one die and each column represents a combination.
* Finally, the modified combinations are printed to the console, showing the values for each die after the modification process.