SRI VENKATESWARA COLLEGE OF ENGINEERING, SRIPERUMBUDUR

NAME	Harini Padma M (Reg No:2127210801033)
DEPARTMENT	B.Tech (Information Technology)

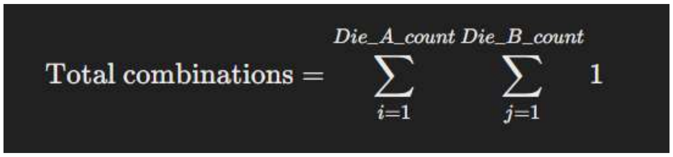
LOGIC AND SOLUTION

PART A:

(i) How many total combinations are possible? Show the math along with the code!

The logic and mathematics behind a code is that, it calculates the number of possible combinations using both the additive and multiplicative principles. The code iterates through all possible combinations of two dice, Die A and Die B, and calculates the total number of combinations using both principles.

Additive Principle:
The total combinations are obtained by summing up the number of choices at each step, iterating through all possible combinations of Die A and Die B.



$$\text{Total combinations} = \sum_{i=1}^{Die\_A\_count} \sum_{j=1}^{Die\_B\_count} 1$$

Where Die_A_count and Die_B_count represents the number if faces of A and B.

Multiplicative principle:
The total combinations are obtained by multiplying the number of choices at each step, using the formula Die_A_count x Die_B_count

Both principles yield the same result for total combinations. The additive principle focuses on summing choices, while the multiplicative principle focuses on multiplying choices. The code demonstrates both principles' equivalence.

(ii) 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!

The total number of combinations can be calculated using multiplicative principle i.e. Die_A_count x Die_B_count. The code uses two nested loops to iterate through all possible combinations of faces on Die A and Die B. The outer loop iterates over the faces of Die A (i), and the inner loop iterates over the faces of Die B (j).

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

The code essentially simulates rolling two six-sided dice and counts the occurrences of each possible sum. It then calculates the probability of each sum occurring by dividing the count of occurrences by the total number of outcomes. The mathematical logic involves enumeration and probability calculation for all possible outcomes of rolling two six-sided dice, providing insights into the distribution of sums and their probabilities.

We know, the probability of an event = Favorable outcomes / Total outcomes.
In a case of two dices,
Total outcomes = 6 × 6 = 36
Let's consider a sum of 6,
Ways of obtaining a sum of 6 are:
(1,5), (5,1), (2,4), (4,2) and (3,3). Thus, there are 5 ways in which 6 can be obtained using two dices.
Therefore, required probability (P),
P(X=6) = 5/ 36

PART B

The Doomed dice problem – solution:

The provided Python code defines functions for generating combinations of dice rolls, calculating the probability distribution of sums of two dice, and transforming two initial sets of dice rolls into new combinations that match a given probability distribution. The `diceA` and `diceB` functions generate combinations of dice rolls recursively using different strategies. The `probSum` function computes the probability distribution of the sum of two dice rolls. The `transform` function attempts to find new combinations of dice rolls that match the probability distribution of the input dice rolls by iterating through all possible combinations and comparing their probability distributions. If a match is found, it prints the new combinations; otherwise, it prints a message indicating no matching combinations were found. Finally, the `main` function initializes two sets of dice rolls, invokes the `transform` function, and prints the result.

Github link: https://github.com/Harini1403/Doomed_dice_chllenge