3. Calculate the Probability of all Possible Sums occurring among the number of combinations from (2). Example: P(Sum = 2) = 1/X as there is only one combination possible to obtain Sum = 2. Die A = Die B = 1.

Explanation:

The output of the code will display the probability of obtaining each possible sum when rolling both dice together. The probabilities are calculated based on the count of occurrences for each sum and the total number of possible outcomes.

1.sides = 6: This variable represents the number of sides on each die. In this case, it's a standard six-sided die.

2.total = sides \*\* 2: The total number of possible outcomes when rolling both dice is calculated by raising the number of sides to the power of 2. This is because each die is independent, so the total number of outcomes is the product of the number of outcomes on each die.

3.d = {}: This is an empty dictionary that will be used to store the count of occurrences for each sum.

4.The nested loops iterate through all possible combinations of outcomes for Die A and Die B. The loop variables i and j represent the outcomes of the two dice.

5.sums = i + j: Calculates the sum of the outcomes of Die A and Die B.

6.d[sums] = d.get(sums, 0) + 1: Updates the count for the current sum in the dictionary d. If the sum is not already in the dictionary, it initializes the count to 0 before incrementing it.

7.After the loops, the code iterates through the dictionary items and calculates the probability for each sum.

8.ans = count / total: Calculates the probability by dividing the count of occurrences for a sum by the total number of possible outcomes.

9.frac = str(count) + "/" + str(total): Formats the count and total as a string in the form "count/total".

10.print(f"P(Sum = {value}) = {frac, ans}"): Prints the probability for each sum along with the count and total.

Code and output:

