**SECURIN ASSESSMENT**

**Github Link:**

https://github.com/abinayass/Assessment

**Collab Link(Code):** https://colab.research.google.com/drive/1eINf02FARWmX2P-7a8z5Q44yx\_mmrDii?usp=sharing

**PART – A**

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

1. Logic to the Given Statement:
   * + When we roll two dice, The total number of combinations can be obtained by multiplying the number of possible outcomes i.e., 1,2,3,4,5,6, of each dice.
     + Therefore the total combinations would be 6×6=36.
2. Explain how did you come up with the solution:

* Calculated the total combinations by using the multiplication operation of the number of faces (6) on each dice

**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!**

**Hint: A 6 x 6 Matrix**

1. Logic to the Given Statement:

* The distribution of all possible combinations is represented as a 6x6 matrix and each cell is a combination obtained by rolling Die A and Die B together
* Iterate over each combination obtained by rolling Die A and Die B
* Calculate their sum

1. Explain how did you come up with the solution:

* Initialized an empty NumPy array with shape (6, 6) to represent the distribution matrix.
* Iterated over each possible value of Die A and Die B, calculated their sum, and filled in the corresponding cell in the matrix with the sum value.

**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**

1. Logic to the Given Statement:

* Since it is asked to calculate the probability of each possible sum occurring, iterate over all the cells in the distribution matrix
* Then, Count the number of occurrences of each sum.
* And divide the count of occurrences by the total number of combinations to obtain the probability of each sum

1. Explain how did you come up with the solution:

* Iterated over each possible sum from 2 to 12.(since it is for the combinations from (2))
* Counted the occurrences of that sum in the distribution matrix.
* Then, divided the count by the total number of combinations to obtain the probability of that sum occurring.
* Finally, printed the probability of each sum with limited decimal places.

**PART – B**

**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**

1. Logic to the Given Statement:

The original probabilities represent the probabilities of each sum occurring with the original dice values for Die A and Die B.

After undooming the dice, the new probabilities represent the probabilities of each sum occurring with the new combinations of dice faces for Die A and Die B, ensuring that the overall probability distribution of sums remains the same as the original probabilities by obeying the Loki’s conditions

1. Explain how did you come up with the solution:

This code adjusts the values of Die A and Die B such that Die A has spots not exceeding 4 per face while maintaining the same distribution. Die B's spots are scaled based on the ratio of the total spots of new\_Die A and Die B, ensuring the probabilities remain unchanged. The modified values of Die A and Die B are then printed.