Securin - Assessment - IIITK - Virothi Venkat Puneeth

Logic to the problem statement – **Part A**  
  
**Problem Statement:**

* We have two six-sided dice, DieA and DieB.
* If we roll both dice together and get a sum.
* Our task is to analyze the possible outcomes and probabilities.

Example: If DieA shows 6 and DieB shows 3, the sum is calculated as 6 + 3 = 9.

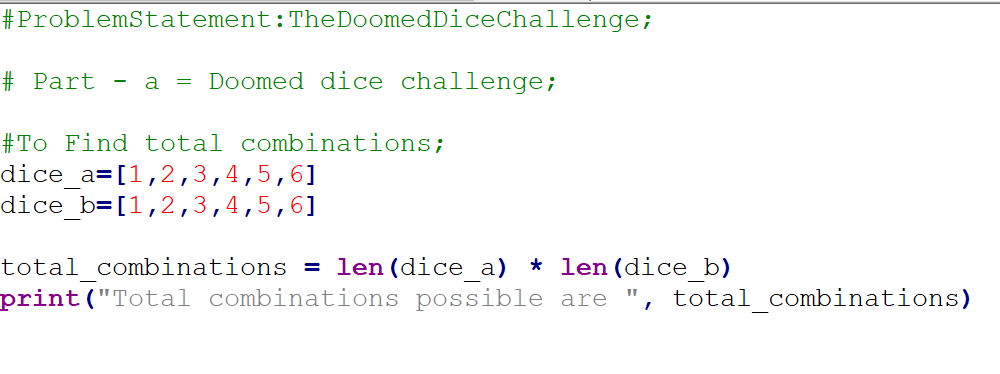
Now, let's go through each part of the problem with a talking example:

**1. Total Combinations:**

You have two dice, each with 6 faces.

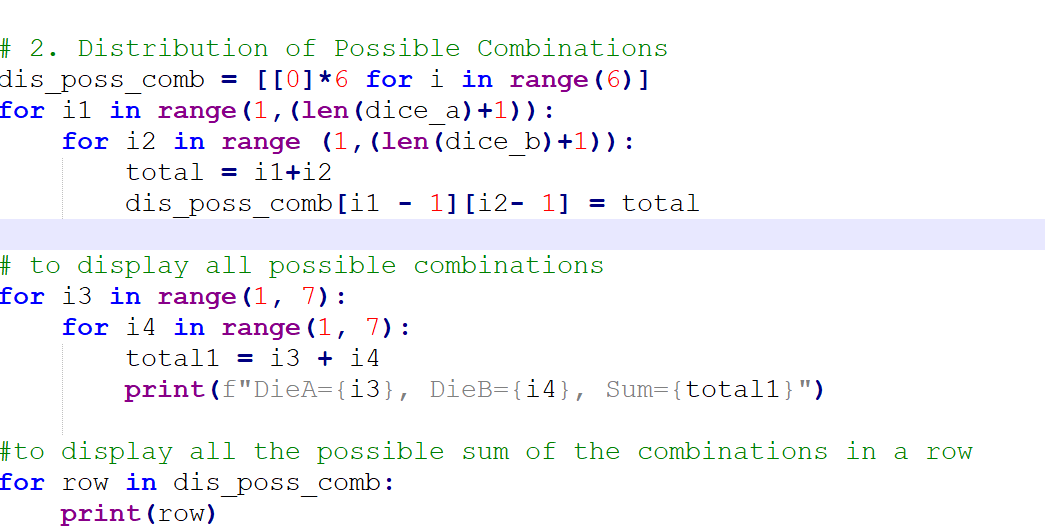
To find the total combinations, you multiply the number of faces on DieA by the number of faces on DieB (6 \* 6).

For our example, there are 36 total combinations possible.  
  
The total number of combinations is calculated by squaring the number of possibilities a single die has. Since each die has 6 sides, squaring this number (6 \* 6) gives the total combinations when rolling two dice.



**2. Distribution of Possible Combinations:**

* Create a 6x6 matrix to represent all possible combinations.
* Iterate through each face of DieA and DieB, calculate the sum, and fill the matrix.
* Display all possible combinations.
* For instance, you might have combinations like DieA=3, DieB=4, Sum=7.



**3. Probability of Sums:**

* Find the minimum and maximum sums from the matrix.
* Iterate through each possible sum and calculate its probability.
* Probability is the ratio of the number of times a sum occurs to the total number of combinations.
* For example, if the sum is 8, and it occurs 5 times out of 36 combinations, the probability is 5/36.

Logic to the problem statement – **Part B**

**To Restore Original Probability Distribution after Loki's Mischief**

Problem Statement (Part-B):

The problem is to undoom the dice that Loki has altered. Loki removed all spots from the dice and we need to reattach them under certain conditions:

● DieAcannothavemorethan4Spotsonaface.

● DieAmayhavemultiplefaceswiththesamenumberofspots.

● DieBcanhaveasmanyspotsonafaceasnecessaryi.e.evenmorethan6.

Butinordertoplayyourgame,theprobabilityofobtainingtheSumsmustremainthe

same!

Soif youcouldonlyrollP(Sum=2)=1/X,thenewdicemusthavethespotsreattached

suchthatthoseprobabilities are notchanged.  
  
**Mathematically way to approach:**

1. **Identify the Range of Possible Sums:**

Determine the potential sum values, ranging from 2 to 12, considering the minimum and maximum face values of the dice from previous solutions(part -A).

1. **Set Fixed Values:**

Ensure a sum of 2 is possible by placing the value 1 in both Die A and Die B.

To enable a sum of 12, place 4 in Die A and 8 in Die B, taking into account Die A's maximum value of 4.

1. **Complete the Remaining Spots:**

For Die A: Utilize only values 2 and 3 to fill the remaining spots, meeting probability constraints. The correct combination is 2, 2, 3, 3.

For Die B: Select values less than 8 to avoid sums exceeding 12. The sole combination that satisfies this condition is 6, 5, 4, 3.

1. **Final Assignments:**

The assigned values for the dice are as follows:

Die A: {1, 2, 2, 3, 3,4}

Die B: {1, 3, 4, 5, 6, 8}

This approach ensures that the dice configurations cover all possible sums between 2 and 12. Fixed values are strategically placed, and the remaining spots are filled while adhering to specific constraints, ensuring the desired probabilities are maintained.

**Explanation of logic that I used in my code:  
find\_dice\_combinations Function:**

This function generates all possible 6-sided dice combinations using recursion.

It starts from a given start\_value and appends values to the temp list until it reaches a length of 6.

The combinations are stored in the combinations list.

**calculate\_sum\_probabilities Function:**

This function calculates the probabilities for sums of two dice.

It uses nested loops to iterate through each combination of values from dice1 and dice2, calculating the sum of the two values.

The sum frequencies are stored in a dictionary called temp\_sums.

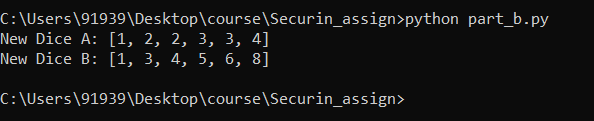
**undoom\_dice Function:**

This function attempts to find transformed dice with matching probabilities to the original sums.

It uses two sets of dice combinations, diceA and diceB, generated by the find\_dice\_combinations function.

It iterates through all combinations of diceA and diceB and calculates the sum probabilities using calculate\_sum\_probabilities.

If it finds a match with the original sums, it returns the combination of dice1, dice2, and the temp\_sums.



Access to Codes – GitHub link - <https://github.com/Puneh222/securin_assign>