**SRI VENKATESWARA COLLEGE OF ENGINEERING**

NAME – BHASKAR.G REGISTRATION NUMBER – 2127210801019

DEPARTMENT – B.TECH INFORMATION TECHNOLOGY

**SOLUTION**

**PART-A**

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

To calculate the combinations there are two ways such as additive and multiplicative principles.

**Additive Principle:**

The logic is that calculate the total combinations by considering each outcome on one die paired with each outcome on the other die.if we have a standard six-sided die, we can roll it and get any number from 1 to 6. If we roll it again, we can get another number from 1 to 6. To find the total combinations, we simply add up all possible pairs of outcomes.

**For Example:**

Since there are 6 outcomes on the first die and 6 outcomes on the second die, you add up all possible pairs: 6+6+6+6+6+6=36

**Multiplicative Principle:**

The logic is that each die has six sides, so there are 6 possible outcomes for each die. Therefore, the total number of combinations is 6×6=36.

1. **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 possible combinations can be calculated by using multiplicative

principle i.e. dieA x dieB. 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, and the inner loop

iterates over the faces of Die B and prints all the possibilities.

1. **Calculate the Probability of all Possible Sums occurring among the**

**number of combinations from (2).**

The logic is that to calculate the probabilities of all possible sums that can occur when two six-sided dice are rolled.Firstly, it constructs a distribution representing the sums resulting from all combinations of face values on the two dice.Then, it computes the probability of each sum by counting the occurrences of each sum in the distribution and dividing by the total number of combinations. Finally, it displays the calculated probabilities for each sum from 2 to 12

**Example:**

* Consider rolling two six-sided dice.
* The code calculates the likelihood of obtaining each possible sum, ranging from 2 to 12.
* For instance, when both dice show a face value of 1, the sum is 2.
* By iterating through all potential combinations of face values on the two dice and tallying the occurrences of each sum, the code determines the probability of each outcome.
* Finally, it presents the probabilities, offering insights into the relative likelihoods of different sums when rolling the dice.

**PART-B**

The primary objective of the solution is to model the transformation process of two dice, named **Die\_A** and **Die\_B**, while accurately computing the probabilities associated with their outcomes. The transformation includes exploring all possible combinations of dice rolls and verifying whether the transformed dice maintain the original probabilities of the outcomes.

**Solution Overview:**

The solution comprises several key components:

1. **Initialization:**
   * Initialization of data structures such as ArrayLists and HashMaps to store dice combinations and their associated probabilities.
2. **Original Probabilities Calculation:**
   * Iteration through all possible combinations of rolls of the original dice (**Die\_A** and **Die\_B**).
   * Calculation of the sums of the two dice and updating a HashMap with the counts of each sum.
   * Computation of probabilities for each sum based on the counts obtained.
3. **Transformation of Dice:**
   * Exploration of all possible combinations of rolls for both **Die\_A** and **Die\_B** after the transformation using recursive methods.
   * Generation of transformed dice combinations while adhering to transformation rules.
4. **Verification of Probabilities:**
   * Calculation of probabilities for each sum occurring with the transformed dice combinations.
   * Comparison of these probabilities with the original probabilities to verify if the transformation preserves the original probabilities.
5. **Result Presentation:**
   * Printing of the transformed dice combinations and their associated probabilities if they match the original probabilities.