Securin The Doomed Dice Challenge

Solution by A. Ahamed Shabith

Part A

1. To find the total combinations possible when rolling both DieA and DieB together, we'll simply multiply the number of faces on each die. Since both are six-sided dice, the total combinations would be 6×6=36.
   The formula "Combinations = number of outcomes of first die * number of outcomes of second die" is further simplified in the code as we consider the faces of the dice rolled as common. Thus, we raise the number of faces on each die to the power of the number of dice rolled, which gives us the total combinations possible.
   Code:

```
public static void combinations_possible(int dice,int n){
//n represents the number of dice
   int x= (int) Math.pow(dice,n);
   System.out.println();
   System.out.println("Total combinations possible with "+n+" dice =
"+x);
}
```

2. We'll iterate through all possible combinations of DieA and DieB, calculate their sums, and create a distribution matrix to represent the frequency of each sum.
3. We'll calculate the probability of each possible sum occurring among the number of combinations from the distribution.

   Formula for Probability,
   Probability = (No. of favorable outcomes / Total outcomes in sample space)

   Code:

```
public static void Probability(int[] arr1, int[] arr2) {
   System.out.println("Probability distribution for all possible sums:");
   int[] sumCount = new int[13];
   for (int num1 : arr1) {
      for (int num2 : arr2) {
         int sum = num1 + num2;
         sumCount[sum]++;
```

```
            }
        }
        double totalCombinations = arr1.length * arr2.length;
        for (int p = 2; p <= 12; p++) {
            double probability = (double) sumCount[p] / totalCombinations;
            System.out.println("P(Sum = " + p + ") : " + probability);
        }
    }
```

Output:

```
Total combinations possible with 2 dice = 36

Displaying all possible combinations when 2 die are rolled together:
-----------------------------------
| 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 |
-----------------------------------
| 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 |
-----------------------------------
| 3,1 | 3,2 | 3,3 | 3,4 | 3,5 | 3,6 |
-----------------------------------
| 4,1 | 4,2 | 4,3 | 4,4 | 4,5 | 4,6 |
-----------------------------------
| 5,1 | 5,2 | 5,3 | 5,4 | 5,5 | 5,6 |
-----------------------------------
| 6,1 | 6,2 | 6,3 | 6,4 | 6,5 | 6,6 |
-----------------------------------
```

```
Sum Distribution Matrix of 2 dice:

-------------------------
| 2 | 3 | 4 | 5 | 6 | 7 |
-------------------------
| 3 | 4 | 5 | 6 | 7 | 8 |
-------------------------
| 4 | 5 | 6 | 7 | 8 | 9 |
-------------------------
| 5 | 6 | 7 | 8 | 9 | 10 |
-------------------------
| 6 | 7 | 8 | 9 | 10 | 11 |
-------------------------
| 7 | 8 | 9 | 10 | 11 | 12 |
-------------------------
```

```
Probability distribution for all possible sums:
P(Sum = 2) : 0.027777777777777776
P(Sum = 3) : 0.05555555555555555
P(Sum = 4) : 0.08333333333333333
P(Sum = 5) : 0.1111111111111111
P(Sum = 6) : 0.1388888888888889
P(Sum = 7) : 0.16666666666666666
P(Sum = 8) : 0.1388888888888889
P(Sum = 9) : 0.1111111111111111
P(Sum = 10) : 0.08333333333333333
P(Sum = 11) : 0.05555555555555555
P(Sum = 12) : 0.027777777777777776

Process finished with exit code 0
```

Part B

The solution to undoom the dice is achieved by finding all possible combinations of their faces and comparing the occurrence of their sums with respect to the original dice. When the occurrence of the sums matches that of the original dice, the solution is found.

The faces of the Undoomed dice are:

New_Dice A : [1, 2, 2, 3, 3, 4]

New_Dice B : [1, 3, 4, 5, 6, 8]

The algorithm is as follows:

1. Initialization:

Define two arrays representing the faces of diceA and diceB, each containing values from 1 to 6.

Call the undoom_dice function with these arrays as arguments.

2. Undooming the Dice (undoom_dice function):

Print the original faces of diceA and diceB.

Calculate the probability distribution of the sums of all possible combinations of the faces of diceA and diceB.

Reset the faces of diceA and diceB to all zeros.

Print the reset (doomed) faces of diceA and diceB.

Define new sets of faces for diceA and diceB (Dice1 and Dice2, respectively).

Generate all possible combinations of faces for diceA and diceB using the Generate_DiceA and Generate_DiceB functions.

Iterate over all combinations of diceA and diceB:

Calculate the probability distribution for each possible sum of faces.

If the distribution matches the original probability distribution, update the faces of diceA and diceB and print them.

Break the loop if the undooming process is successful.

Print the probability distribution of the transformed dice.

3. Dice Transformation:

The Generate_DiceA and Generate_DiceB functions generate all possible combinations of faces for diceA and diceB, respectively, using recursion.

The ProbabilityMap function calculates the probability distribution of the sums of all possible combinations of faces for diceA and diceB.

The Probability function calculates the probability distribution for the sums of faces of the dice.

4. Utility Functions:

The convertor function converts a list of integers to an array of integers.

The DiceA and DiceB functions recursively generate all possible combinations of faces for new_diceA and new_diceB, respectively.

Output:

```
Dice A : [1, 2, 3, 4, 5, 6]
Dice B : [1, 2, 3, 4, 5, 6]

Probability Distribution of Original dice:

P(Sum = 2) : 0.027777777777777776
P(Sum = 3) : 0.05555555555555555
P(Sum = 4) : 0.08333333333333333
P(Sum = 5) : 0.1111111111111111
P(Sum = 6) : 0.1388888888888889
P(Sum = 7) : 0.16666666666666666
P(Sum = 8) : 0.1388888888888889
P(Sum = 9) : 0.1111111111111111
P(Sum = 10) : 0.08333333333333333
P(Sum = 11) : 0.05555555555555555
P(Sum = 12) : 0.027777777777777776

Doomed dice A : [0, 0, 0, 0, 0, 0]
Doomed dice B : [0, 0, 0, 0, 0, 0]
```

```
The faces of the Undoomed dice are:
New_Dice A : [1, 2, 2, 3, 3, 4]
New_Dice B : [1, 3, 4, 5, 6, 8]

Probability Distribution of the transformed dice:

P(Sum = 2) : 0.027777777777777776
P(Sum = 3) : 0.05555555555555555
P(Sum = 4) : 0.08333333333333333
P(Sum = 5) : 0.1111111111111111
P(Sum = 6) : 0.1388888888888889
P(Sum = 7) : 0.16666666666666666
P(Sum = 8) : 0.1388888888888889
P(Sum = 9) : 0.1111111111111111
P(Sum = 10) : 0.08333333333333333
P(Sum = 11) : 0.05555555555555555
P(Sum = 12) : 0.027777777777777776

Process finished with exit code 0
```