**Problem Statement: The Doomed Dice Challenge**

**Part-A (15-20 Minutes):**

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

**Code:**

def doomed\_dice\_challenge ():

die\_a = [1, 2, 3, 4, 5, 6]

die\_b = [1, 2, 3, 4, 5, 6]

total combinations = 0

for face\_a in die\_a:

for face\_b in die\_b:

total\_sum = face\_a + face\_b

print(f"Die A = {face\_a}, Die B = {face\_b}. Sum = {total\_sum}")

total\_combinations += 1

print(f"Total Combinations: {total\_combinations}")

doomed\_dice\_challenge()

**Output:**

Die A = 1, Die B = 1. Sum = 2

Die A = 1, Die B = 2. Sum = 3

Die A = 1, Die B = 3. Sum = 4

Die A = 1, Die B = 4. Sum = 5

Die A = 1, Die B = 5. Sum = 6

Die A = 1, Die B = 6. Sum = 7

Die A = 2, Die B = 1. Sum = 3

Die A = 2, Die B = 2. Sum = 4

Die A = 2, Die B = 3. Sum = 5

Die A = 2, Die B = 4. Sum = 6

Die A = 2, Die B = 5. Sum = 7

Die A = 2, Die B = 6. Sum = 8

Die A = 3, Die B = 1. Sum = 4

Die A = 3, Die B = 2. Sum = 5

Die A = 3, Die B = 3. Sum = 6

Die A = 3, Die B = 4. Sum = 7

Die A = 3, Die B = 5. Sum = 8

Die A = 3, Die B = 6. Sum = 9

Die A = 4, Die B = 1. Sum = 5

Die A = 4, Die B = 2. Sum = 6

Die A = 4, Die B = 3. Sum = 7

Die A = 4, Die B = 4. Sum = 8

Die A = 4, Die B = 5. Sum = 9

Die A = 4, Die B = 6. Sum = 10

Die A = 5, Die B = 1. Sum = 6

Die A = 5, Die B = 2. Sum = 7

Die A = 5, Die B = 3. Sum = 8

Die A = 5, Die B = 4. Sum = 9

Die A = 5, Die B = 5. Sum = 10

Die A = 5, Die B = 6. Sum = 11

Die A = 6, Die B = 1. Sum = 7

Die A = 6, Die B = 2. Sum = 8

Die A = 6, Die B = 3. Sum = 9

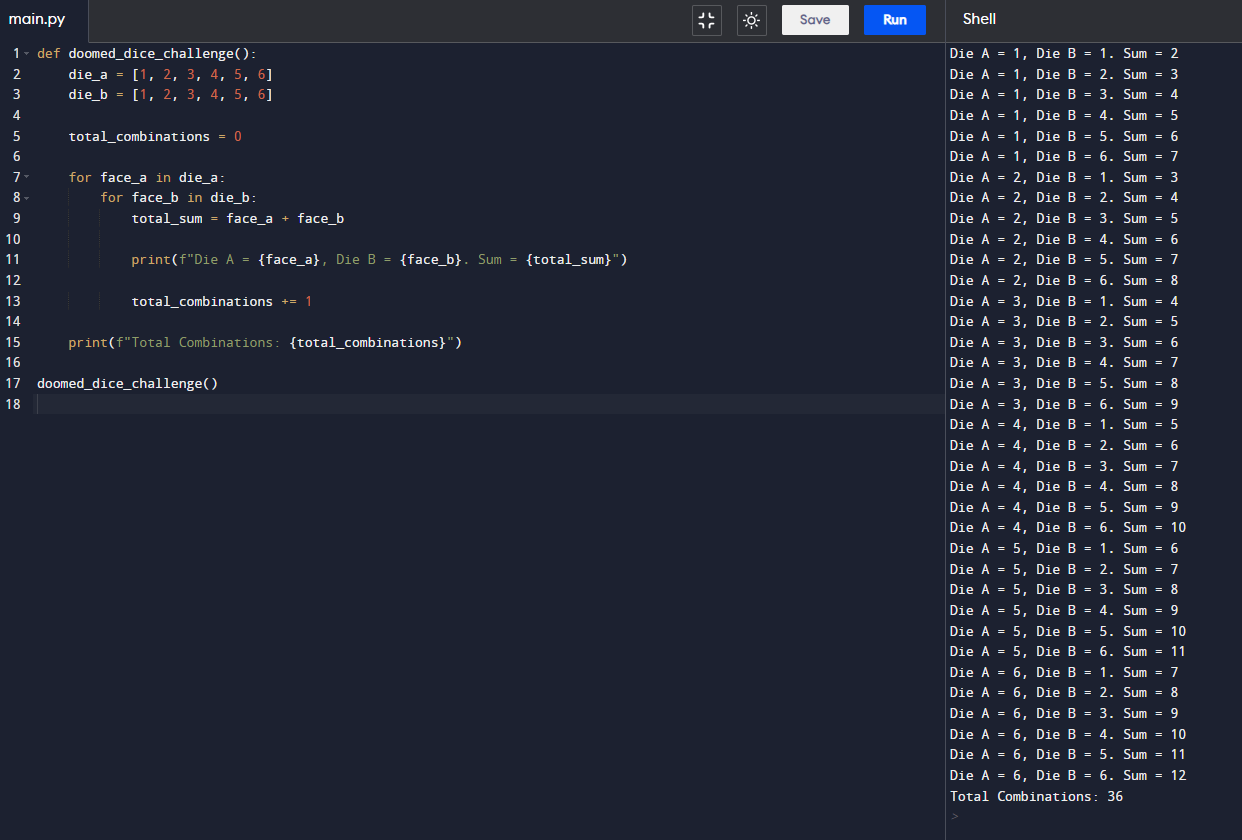
Die A = 6, Die B = 4. Sum = 10

Die A = 6, Die B = 5. Sum = 11

Die A = 6, Die B = 6. Sum = 12

Total Combinations: 36

>



**2. 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!**

**Code:**

def doomed\_dice\_challenge():

die\_a = [1, 2, 3, 4, 5, 6]

die\_b = [1, 2, 3, 4, 5, 6]

distribution\_matrix = [[0] \* 6 for \_ in range(6)]

for face\_a in die\_a:

for face\_b in die\_b:

total\_sum = face\_a + face\_b

distribution\_matrix[face\_a - 1][face\_b - 1] += 1

print(f"Distribution Matrix:")

for row in distribution\_matrix:

print(row)

doomed\_dice\_challenge()

**Output:**

Distribution Matrix:

[1, 1, 1, 1, 1, 1]

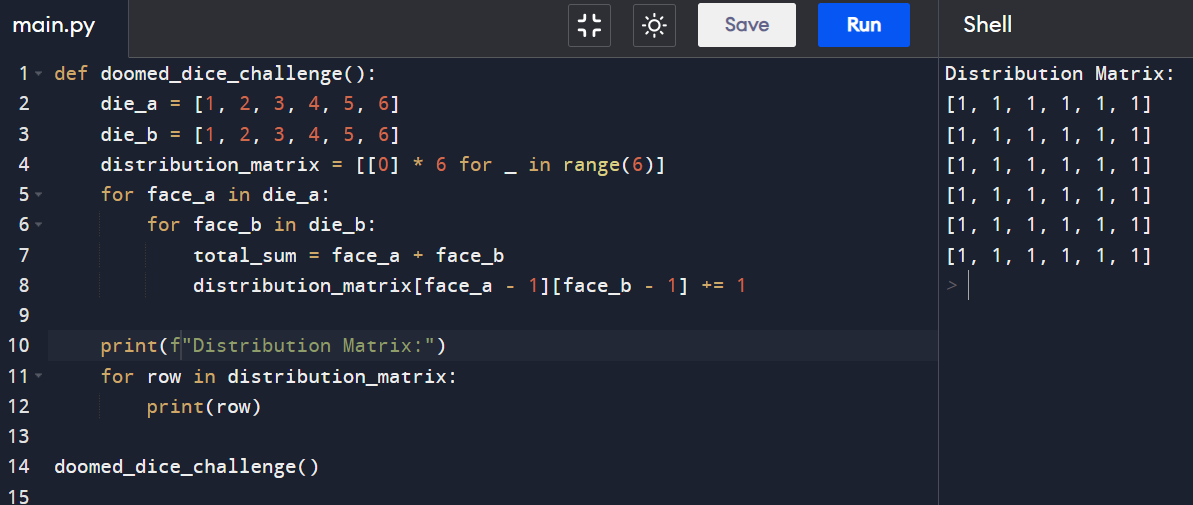
[1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1]>



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.**

**Code:**

def doomed\_dice\_challenge():

die\_a = [1, 2, 3, 4, 5, 6]

die\_b = [1, 2, 3, 4, 5, 6]

distribution\_matrix = [[0] \* 6 for \_ in range(6)]

probabilities = {}

total\_combinations = 0

for face\_a in die\_a:

for face\_b in die\_b:

total\_sum = face\_a + face\_b

distribution\_matrix[face\_a - 1][face\_b - 1] += 1

total\_combinations += 1

if total\_sum not in probabilities:

probabilities[total\_sum] = 1

else:

probabilities[total\_sum] += 1

print(f"Probabilities:")

for key, value in probabilities.items():

probability = value / total\_combinations

print(f"P(Sum = {key}) = {1/total\_combinations} (1/{total\_combinations})")

doomed\_dice\_challenge()

**Output:**

Probabilities:

P(Sum = 2) = 0.027777777777777776 (1/36)

P(Sum = 3) = 0.027777777777777776 (1/36)

P(Sum = 4) = 0.027777777777777776 (1/36)

P(Sum = 5) = 0.027777777777777776 (1/36)

P(Sum = 6) = 0.027777777777777776 (1/36)

P(Sum = 7) = 0.027777777777777776 (1/36)

P(Sum = 8) = 0.027777777777777776 (1/36)

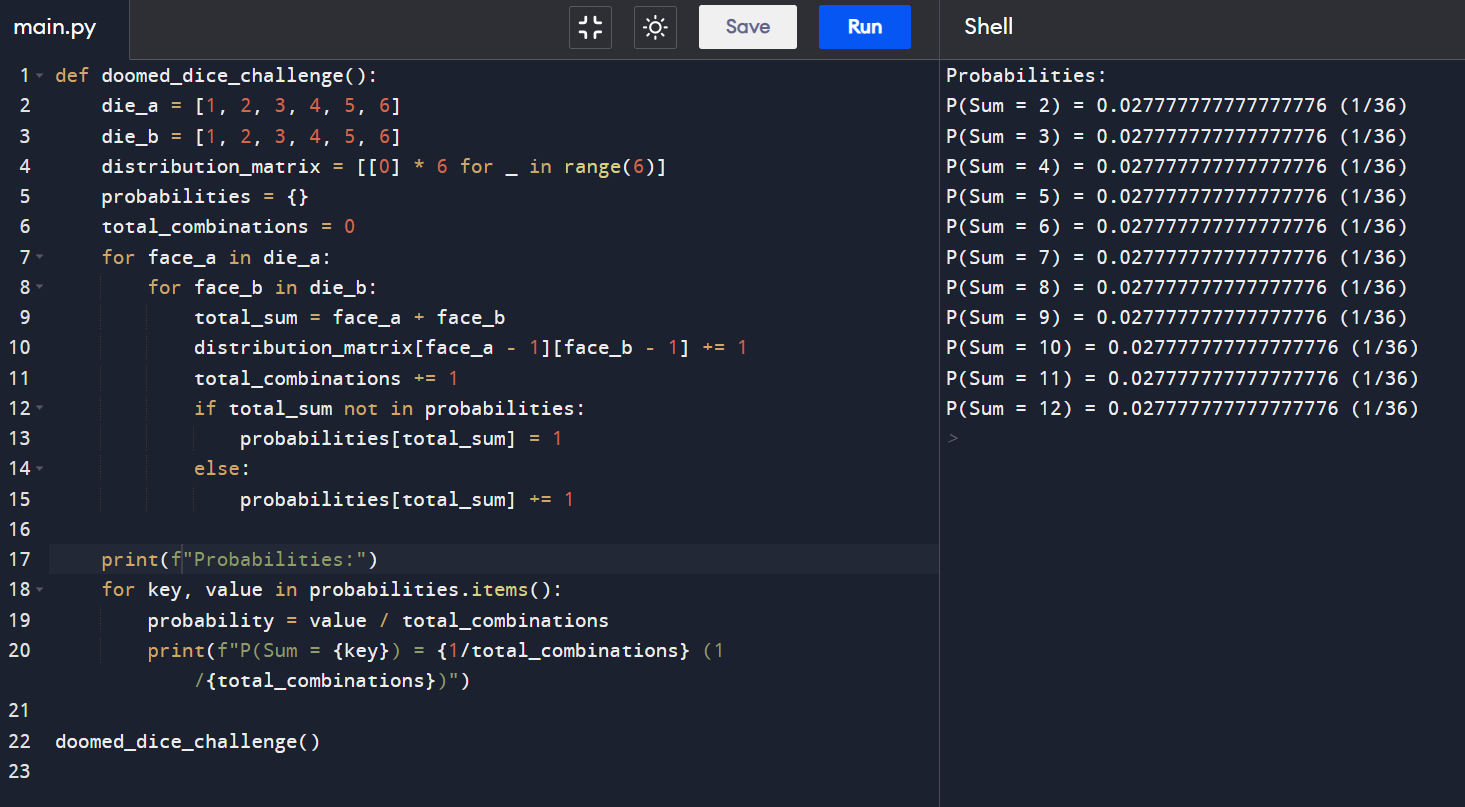
P(Sum = 9) = 0.027777777777777776 (1/36)

P(Sum = 10) = 0.027777777777777776 (1/36)

P(Sum = 11) = 0.027777777777777776 (1/36)

P(Sum = 12) = 0.027777777777777776 (1/36)

>



Part-B (25-30)

**Code**:

def undoom\_dice(die\_a, die\_b):

new\_die\_a = [min(4, x) for x in die\_a]

new\_die\_b = die\_b

return new\_die\_a, new\_die\_b

die\_a = [1, 2, 3, 4, 5, 6]

die\_b = [1, 2, 3, 4, 5, 6]

new\_die\_a, new\_die\_b = undoom\_dice(die\_a, die\_b)

print(f"\nInitial Die A:", die\_a)

print(f"Initial Die B:", die\_b)

print(f"\nNew Die A:", new\_die\_a)

print(f"New Die B:", new\_die\_b)

**Output:**

Initial Die A: [1, 2, 3, 4, 5, 6]

Initial Die B: [1, 2, 3, 4, 5, 6]

New Die A: [1, 2, 3, 4, 4, 4]

New Die B: [1, 2, 3, 4, 5, 6]

>

