Code and Explanation

# Define the number of sides for each die

num\_sides\_die\_a = 6

num\_sides\_die\_b = 6

total\_combinations = num\_sides\_die\_a \* num\_sides\_die\_b

print("Total Combinations:", total\_combinations)

# Create an empty matrix to store combinations

combinations\_matrix = [[0 for \_ in range(num\_sides\_die\_b)] for \_ in range(num\_sides\_die\_a)]

# Loop through each element of the matrix

for i in range(num\_sides\_die\_a):

for j in range(num\_sides\_die\_b):

# Calculate the sum of the dice rolls for the current combination

combinations\_matrix[i][j] = i + j + 2

# Print the combinations matrix

print("Combinations Matrix:")

for row in combinations\_matrix:

print(row)

Step 1: Define the number of sides for each die

The code defines two variables, num\_sides\_die\_a and num\_sides\_die\_b, to represent the number of sides on each die. In this case, both dice have 6 sides.

Step 2: Calculate the total number of combinations

The code then multiplies the number of sides on each die to find the total number of possible combinations. For example, with two 6-sided dice, there are 6 \* 6 = 36 possible combinations.

Step 3: Create a matrix to store the combinations

The code creates a two-dimensional matrix, combinations\_matrix, to store the sum of each possible dice roll combination. The matrix has dimensions num\_sides\_die\_a (rows) by num\_sides\_die\_b (columns).

Step 4: Loop through each element of the matrix

The code uses nested loops to iterate through each element of the matrix. For each element, the code:

* Calculates the row index (i) and column index (j).
* Calculates the sum of the current dice roll combination by adding i and j.
* Adds 2 to the sum to account for the minimum possible roll (1 on each die).
* Stores the sum in the corresponding element of the combinations\_matrix.

Step 5: Print the combinations matrix

Finally, the code prints the combinations\_matrix. Each row of the matrix represents the possible sums for a given value of the first die (row index). Each column of the matrix represents the possible sums for a given value of the second die (column index).

**Math behind the code:**

The math behind the code is quite simple. The sum of two dice rolls is simply the sum of the numbers rolled on each die. However, the code needs to account for the minimum possible roll (1 on each die) being 2, not 0. Therefore, 2 is added to each sum in the combinations\_matrix.

For example, consider the element in the first row and first column of the matrix. This element represents the sum of rolling a 1 on the first die and a 1 on the second die. The sum of these rolls is 1 + 1 = 2. However, since the minimum possible roll is 2, we add 2 to get 2 + 2 = 4. Therefore, the element in the first row and first column of the matrix is 4.