**Part A:**

total\_combinations = 6 \* 6

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

# Initialize an empty 6x6 matrix

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

# Fill in the matrix with sums

for i in range(6):

for j in range(6):

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

# Display the matrix

for row in combinations\_matrix:

print(row)

# Initialize a dictionary to store probabilities

sum\_probabilities = {}

# Calculate probabilities

for i in range(2, 13):

count = sum(row.count(i) for row in combinations\_matrix)

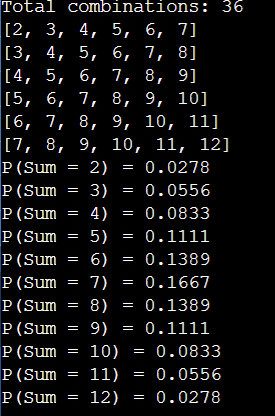
probability = count / total\_combinations

sum\_probabilities[i] = probability

# Display probabilities

for sum\_value, prob in sum\_probabilities.items():

print(f"P(Sum = {sum\_value}) = {prob:.4f}")  
  
  
 **Output:**

****

**Explanation**

* total\_combinations = 6 \* 6: This line calculates the total number of combinations (i.e., possible outcomes) of rolling two six-sided dice. There are 6 sides on each die, so there are 6 \* 6 = 36 possible combinations.
* print(f"Total combinations: {total\_combinations}"): This line prints the total number of combinations.
* combinations\_matrix = [[0] \* 6 for \_ in range(6)]: This line initializes an empty 6x6 matrix using a list comprehension. It creates a list of 6 zeros for each row, and there are 6 rows in total.
* for i in range(6):: This line starts a loop that iterates over the range of numbers from 0 to 5 (inclusive). The variable i represents the current number in the range.
* for j in range(6):: This line starts a nested loop that also iterates over the range of numbers from 0 to 5 (inclusive). The variable j represents the current number in the range.
* combinations\_matrix[i][j] = i + j + 2: This line calculates the sum of the current numbers i and j (which represent the numbers rolled on the two dice), adds 2 to account for the fact that the dice have 6 sides (i.e., the lowest possible sum is 2), and stores the result in the corresponding cell of the combinations\_matrix.
* for row in combinations\_matrix:: This line starts a loop that iterates over each row in the combinations\_matrix.
* print(row): This line prints the current row of the matrix.
* sum\_probabilities = {}: This line initializes an empty dictionary to store the probabilities of each possible sum.
* for i in range(2, 13):: This line starts a loop that iterates over the range of numbers from 2 to 12 (inclusive). The variable i represents the current sum.
* count = sum(row.count(i) for row in combinations\_matrix): This line calculates the number of times the current sum i appears in the combinations\_matrix. It does this by using a generator expression to iterate over each row in the matrix, counting the number of times i appears in that row, and summing up all the counts.
* probability = count / total\_combinations: This line calculates the probability of the current sum i by dividing the count of that sum by the total number of combinations.
* sum\_probabilities[i] = probability: This line stores the probability of the current sum i in the sum\_probabilities dictionary.
* for sum\_value, prob in sum\_probabilities.items():: This line starts a loop that iterates over each sum and its corresponding probability in the sum\_probabilities dictionary. The variables sum\_value and prob represent the current sum and its probability, respectively.
* print(f"P(Sum = {sum\_value}) = {prob:.4f}"): This line prints the probability of the current sum sum\_value. The :.4f part of the string format specifies that the probability should be formatted as a floating-point number with 4 decimal places.