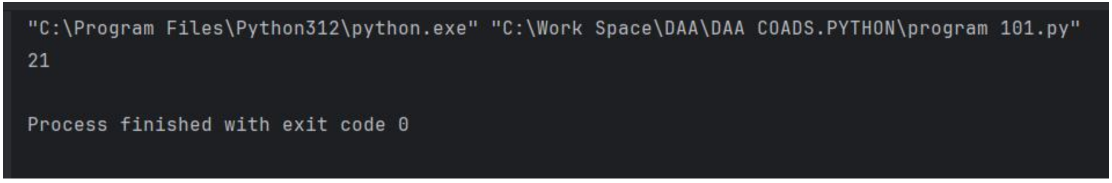
Program 101.Dice throw problem

Program:

```
def num_ways_to_get_sum(n, k, X):
    # Create a 2D dp array with dimensions (n+1) x (X+1)
    dp = [[0] * (X + 1) for _ in range(n + 1)]

    # Base case: There's 1 way to get a sum of 0 with 0 dice
    dp[0][0] = 1

    # Fill the dp array
    for i in range(1, n + 1):  # Loop through the number of dice
        for j in range(1, X + 1):  # Loop through all possible sums
            dp[i][j] = 0  # Initialize current cell
            # Calculate number of ways to get sum j with i dice
            for face in range(1, k + 1):
                if j - face >= 0:
                    dp[i][j] += dp[i - 1][j - face]

    # Return the number of ways to get sum X with n dice
    return dp[n][X]

# Example usage
n = 3  # Number of dice
k = 6  # Number of faces on each die
X = 8  # Target sum
print(num_ways_to_get_sum(n, k, X))  # Output: 21
```

Output:



```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA\DAA COADS.PYTHON\program 101.py"
21

Process finished with exit code 0
```

Time complexity:

O(n.X.k)