4.1 DICE THROW PROBLEM

Question:

You are given the number of sides on a die (num_sides), the number of dice to throw (num_dice), and a target sum (target). Develop a program that utilizes dynamic programming to solve the Dice Throw Problem.

AIM

To design a Python program that calculates the number of ways to achieve a target sum using a given number of dice and sides, using dynamic programming.

ALGORITHM

1. Define a 2D DP table dp[d][t] where d is the number of dice used and t is the current sum.
2. Initialize dp[0][0] = 1 (1 way to get sum 0 with 0 dice).
3. For each dice from 1 to num_dice:
    - For each sum from 1 to target:
    - For each face value from 1 to num_sides:
    - If t - face >= 0, add dp[d-1][t-face] to dp[d][t].
4. Return dp[num_dice][target].

PROGRAM

```
def dice_throw(num_dice, num_sides, target):
    dp = [[0] * (target + 1) for _ in range(num_dice + 1)]
    dp[0][0] = 1
    for i in range(1, num_dice + 1):
        for j in range(1, target + 1):
            for k in range(1, num_sides + 1):
                if j - k >= 0:
                    dp[i][j] += dp[i - 1][j - k]
    return dp[num_dice][target]

num_sides = int(input("Enter number of sides on the die: "))
num_dice = int(input("Enter number of dice: "))
target = int(input("Enter target sum: "))
print(f"Number of ways to reach sum {target}: {dice_throw(num_dice, num_sides, target)}")
```

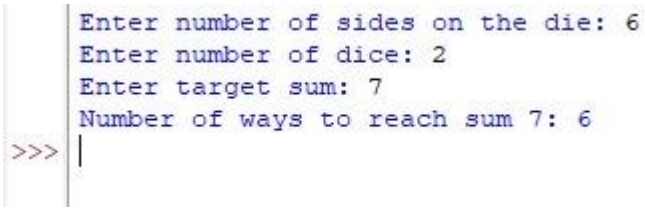
Input:

No. of sides: 6

No. of dice: 2

Target:7

Output:



RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: O(num_dice × target × num_sides)
- Space Complexity: O(num_dice × target)