6.5 TARGET SUM EXPRESSIONS

Question:

You are given an integer array nums and an integer target. You want to build an expression out of nums by adding one of the symbols '+' and '-' before each integer in nums and then concatenate all the integers.

For example, if nums = [2, 1], you can add a '+' before 2 and a '-' before 1 and concatenate them to build the expression '+2-1'. Return the number of different expressions that you can build, which evaluates to target.

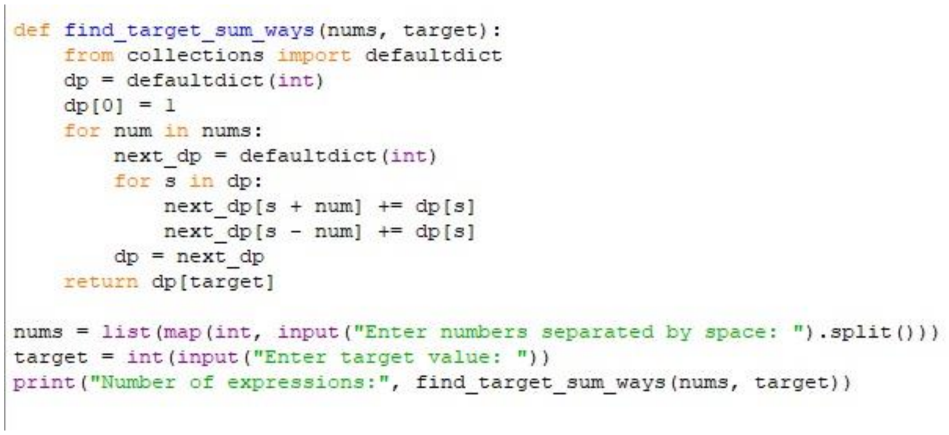
AIM

To implement a program in Python that counts the number of ways to assign '+' and '-' signs to array elements so that their sum equals the target.

ALGORITHM

1. Use Depth First Search (DFS) or Dynamic Programming to explore possible assignments of '+' and '-' to each element.
2. At each index, branch into two recursive calls:
3. - One with '+' applied to the current number.
4. - Another with '-' applied to the current number.
5. Continue until all numbers are processed.
6. If the cumulative sum equals the target, count that as a valid expression.
7. Return the total count of valid expressions.

PROGRAM



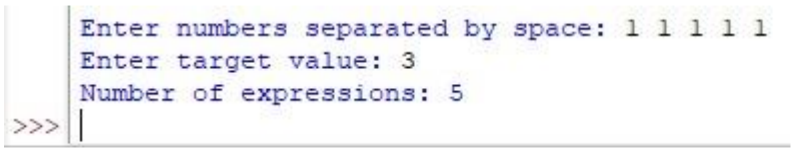
```
def find_target_sum_ways(nums, target):
    from collections import defaultdict
    dp = defaultdict(int)
    dp[0] = 1
    for num in nums:
        next_dp = defaultdict(int)
        for s in dp:
            next_dp[s + num] += dp[s]
            next_dp[s - num] += dp[s]
        dp = next_dp
    return dp[target]

nums = list(map(int, input("Enter numbers separated by space: ").split()))
target = int(input("Enter target value: "))
print("Number of expressions:", find_target_sum_ways(nums, target))
```

Input:

nums = [1,1,1,1,1], target = 3

Output:



```
Enter numbers separated by space: 1 1 1 1 1
Enter target value: 3
Number of expressions: 5
>>> |
```

RESULT:

Thus, the program is successfully executed and verified to count the number of valid target sum expressions.

PERFORMANCE ANALYSIS:

Time Complexity: O(n * sum(nums)) due to memoization where n is the number of elements.

Space Complexity: O(n * sum(nums)) for storing memoization states.