210. 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.

Example 1:
Input: nums = [1,1,1,1,1], target = 3
Output: 5
Explanation: There are 5 ways to assign symbols to make the sum of nums be target 3.
-1 + 1 + 1 + 1 + 1 = 3
+1 - 1 + 1 + 1 + 1 = 3
+1 + 1 - 1 + 1 + 1 = 3
+1 + 1 + 1 - 1 + 1 = 3
+1 + 1 + 1 + 1 - 1 = 3
Example 2:
Input: nums = [1], target = 1
Output: 1

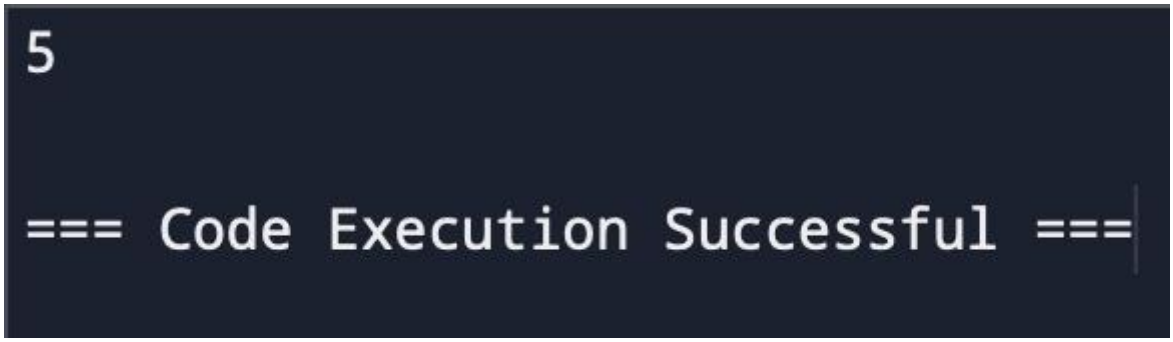
PROGRAM:-

```
from collections import defaultdict

def findTargetSumWays(nums, target):
    dp = defaultdict(int)
    dp[0] = 1
    for num in nums:
        next_dp = defaultdict(int)
        for sum_val, count in dp.items():
            next_dp[sum_val + num] += count
            next_dp[sum_val - num] += count
        dp = next_dp
    return dp[target]

# Example
nums = [1, 1, 1, 1, 1]
target = 3
print(findTargetSumWays(nums, target))
```

OUTPUT:-



TIME COMPLEXITY:-O(N)