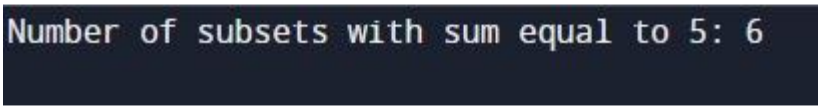
86. Meet in middle technique

AIM:To Find the meet in the middle technique by using divide and conquer method

PROGRAM:

```
def subset_sum(nums, target):
    n = len(nums)
    half = n // 2
    subsets1 = []
    for i in range(1 << half):  # iterate over all 2^half subsets
        subset_sum = sum(nums[j] for j in range(half) if (i & (1 << j)) > 0)
        subsets1.append(subset_sum)
    subsets2 = []
    for i in range(1 << (n - half)):  # iterate over all 2^(n-half) subsets
        subset_sum = sum(nums[half + j] for j in range(n - half) if (i & (1 << j)) > 0)
        subsets2.append(subset_sum)
    subsets2.sort()
    count = 0
    for sum1 in subsets1:
        left, right = 0, len(subsets2) - 1
        while left <= right:
            mid = (left + right) // 2
            if subsets2[mid] + sum1 == target:
                count += 1
                left_count = mid
                while left_count >= 0 and subsets2[left_count] == subsets2[mid]:
                    count += 1
                    left_count -= 1
                right_count = mid + 1
                while right_count < len(subsets2) and subsets2[right_count] == subsets2[mid]:
                    count += 1
                    right_count += 1
                break
            elif subsets2[mid] + sum1 < target:
                left = mid + 1
            else:
                right = mid - 1

    return count
nums = [1, 2, 3, 4, 5]
target = 5
result = subset_sum(nums, target)
print(f"Number of subsets with sum equal to {target}: {result}")
```

OUTPUT: 

```
Number of subsets with sum equal to 5: 6
```

TIME COMPLEXITY: O($n^{2)}$