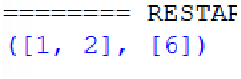
EXERCISE-87 Meet in middle technique

PROGRAM

```
def meet_in_the_middle(nums, target):
    half = len(nums) // 2
    left_half = nums[:half]
    right_half = nums[half:]
    left_subsets = [[]]
    for num in left_half:
        left_subsets += [subset + [num] for subset in left_subsets]
    right_subsets = [[]]
    for num in right_half:
        right_subsets += [subset + [num] for subset in right_subsets]
    left_subsets.sort()
    right_subsets.sort()
    left, right = 0, len(right_subsets) - 1
    while left < len(left_subsets) and right >= 0:
        total = sum(left_subsets[left]) + sum(right_subsets[right])
        if total == target:
            return left_subsets[left], right_subsets[right]
        elif total < target:
            left += 1
        else:
            right -= 1
    return None
nums = [1, 2, 3, 4, 5, 6]
target = 9
result = meet_in_the_middle(nums, target)
print(result)
```

OUTPUT



```
======== RESTAR
([1, 2], [6])
```

TIME COMPLEXITY $O(n\cdot 2n/2)$.