Started on	Tuesday, 8 April 2025, 11:44 AM
State	Finished
Completed on	Tuesday, 8 April 2025, 11:46 AM
Time taken	1 min 55 secs
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

Mark 4.00 out of 4.00

SUBSET SUM PROBLEM

COUNT OF SUBSETS WITH SUM EQUAL TO X

Given an array arr[] of length N and an integer X, the task is to find the number of subsets with a sum equal to X.

Examples:

Input: *arr[] = {1, 2, 3, 3}, X = 6*
Output: *3*
All the possible subsets are {1, 2, 3},
{1, 2, 3} and {3, 3}

Input: *arr[] = {1, 1, 1, 1}, X = 1*
Output: *4*

THE INPUT

1.No of numbers

2.Get the numbers

3.Sum Value

For example:

Input	Result
4 2 4 5 9 15	1
6 3 34 4 12 3 2 7	2

Answer: (penalty regime: 0 %)

Reset answer

```
from itertools import combinations

def count_subsets(numbers, target_sum):
    count = 0
    n = len(numbers)

    # Generate all possible subsets
    for i in range(1, n + 1):  # Subset sizes from 1 to n
        for subset in combinations(numbers, i):
            if sum(subset) == target_sum:
                count += 1

    return count

n = int(input())
numbers = []
for i in range(n):
    value = int(input())
```

```
    numbers.append(value)

target_sum = int(input())
```

	Input	Expected	Got	
✔	4 2 4 5 9 15	1	1	✔
✔	6 10 20 25 50 70 90 80	2	2	✔
✔	5 4 16 5 23 12 9	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 4.00/4.00.

Question **2**

Correct

Mark 3.00 out of 3.00

SUBSET SUM PROBLEM

Given a set of positive integers, and a value sum, determine that the sum of the subset of a given set is equal to the given sum.

Write the program for subset sum problem.

INPUT

1.no of elements

2.Input the given elements

3.Get the target sum

OUTPUT

True , if subset with required sum is found

False , if subset with required sum is not found

For example:

Input	Result
5 4 16 5 23 12 9	4 16 5 23 12 True,subset found

Answer: (penalty regime: 0 %)

Reset answer

```
def SubsetSum(a,i,sum,target,n):
# Write your code here
    if i==n:
        return sum==target
    if sum>target:
        return False
    if sum==target:
        return True
    return SubsetSum(a,i+1,sum,target,n) or SubsetSum(a,i+1,sum+a[i],target,n)


a=[]
size=int(input())
for i in range(size):
    x=int(input())
    a.append(x)

target=int(input())
n=len(a)
if(SubsetSum(a,0,0,target,n)==True):
    for i in range(size):
        print(a[i])
```

	Input	Expected	Got	
✔	5 4 16 5 23 12 9	4 16 5 23 12 True,subset found	4 16 5 23 12 True,subset found	✔

	Input	Expected	Got	
✔	4 1 2 3 4 11	1 2 3 4 False,subset not found	1 2 3 4 False,subset not found	✔
✔	7 10 7 5 18 12 20 15 35	10 7 5 18 12 20 15 True,subset found	10 7 5 18 12 20 15 True,subset found	✔

Passed all tests! ✔

Correct

Marks for this submission: 3.00/3.00.

Question **3**

Correct

Mark 3.00 out of 3.00

SUBSET SUM PROBLEM

We are given a list of n numbers and a number x, the task is to write a python program to find out all possible subsets of the list such that their sum is x.

Examples:

Input: *arr = [2, 4, 5, 9], x = 15*

Output: *[2, 4, 9]*

15 can be obtained by adding 2, 4 and 9 from the given list.

Input : *arr = [10, 20, 25, 50, 70, 90], x = 80*

Output : *[10, 70]*

[10, 20, 50]

80 can be obtained by adding 10 and 70 or by adding 10, 20 and 50 from the given list.

THE INPUT

1.No of numbers

2.Get the numbers

3.Sum Value

For example:

Input	Result
4 2 4 5 9 15	[2, 4, 9]
5 4 16 5 23 12 9	[4, 5]

Answer: (penalty regime: 0 %)

Reset answer

```
from itertools import combinations
def subsetSum(n, arr, x):
    for i in range(n+1):
        for subset in combinations(arr, i):
            if sum(subset) == x:
                print(list(subset))


n=int(input())
arr=[]
for i in range(0,n):
    a=int(input())
    arr.append(a)
x = int(input())

subsetSum(n, arr, x)
```

	Input	Expected	Got	
✔	4 2 4 5 9 15	[2, 4, 9]	[2, 4, 9]	✔
✔	6 10 20 25 50 70 90 80	[10, 70] [10, 20, 50]	[10, 70] [10, 20, 50]	✔
✔	5 4 16 5 23 12 9	[4, 5]	[4, 5]	✔

Passed all tests! ✔

Correct

Marks for this submission: 3.00/3.00.