Author

[Pingal\_T](https://codefights.com/profile/Pingal_T)

https://codefights.com/img/coins_new.png3000

You are given an array of integers ARR and an integerSUM.

Your task is to find the number of subsets of the arrayARR, such that the sum of their elements equals SUM

**Example:**

* SubsetSum([1,2,3,4,5], 5) = 3  
  These subsets are [1,4], [2,3] and [5].
* SubsetSum([1,2,3,4,-5], 0) = 3  
  These subsets are [1,4,-5], [2,3,-5] and [].

Note that a subset can be empty.

* **[input] array.integer ARR**
  + Array of integers, 0 < ARR.length ≤ 15
* **[input] integer SUM**
* **[output] integer**

<https://codefights.com/challenge/RpcMTbHAL4PcdZjdR>

List<List<int>> powerset(List<int> list)

{

List<List<int>> ps = new List<List<int>>();

ps.Add(new List<int>()); // add the empty set

// for every item in the original list

foreach (int item in list)

{

List<List<int>> newPs = new List<List<int>>();

foreach (List<int> subset in ps)

{

// copy all of the current powerset's subsets

newPs.Add(subset);

// plus the subsets appended with the current item

List<int> newSubset = new List<int>(subset);

newSubset.Add(item);

newPs.Add(newSubset);

}

// powerset is now powerset of list.subList(0, list.indexOf(item)+1)

ps = newPs;

}

return ps;

}

int SubsetSum(int[] ARR, int SUM)

{

List<int> lista = new List<int>(ARR);

List<List<int>> comb = powerset(lista);

int ans = 0;

for (int i = 0; i < comb.Count; i++)

{

int sum\_comb = 0;

for (int j = 0; j < comb[i].Count; j++)

{

sum\_comb += comb[i][j];

}

if (sum\_comb == SUM)

{

ans++;

}

}

return ans;

}