**Subset Sum Problem:-**

Given an array **arr[]** of size **N**, check if it can be partitioned into two parts such that the sum of elements in both parts is the same.

**Example 1:**

**Input:** N = 4

arr = {1, 5, 11, 5}

**Output:** YES

**Explaination:**

The two parts are {1, 5, 5} and {11}.

**Example 2:**

**Input:** N = 3

arr = {1, 3, 5}

**Output:** NO

**Explaination:** This array can never be

partitioned into two such parts.

**Your Task:**  
You do not need to read input or print anything. Your task is to complete the function **equalPartition()** which takes the value N and the array as input parameters and returns 1 if the partition is possible. Otherwise, returns 0.

**Expected Time Complexity:** O(N\*sum of elements)  
**Expected Auxiliary Space:** O(N\*sum of elements)

**Constraints:**  
1 ≤ N ≤ 100  
1 ≤ arr[i] ≤ 1000