**Perfect Sum Problem**

**Medium**Accuracy: 47.14% Submissions: 21221 Points: 4

Given an array **arr[]** of integers and an integer **sum**, the task is to count all subsets of the given array with a sum equal to a given **sum**.

Note: Answer can be very large, so, output answer modulo 109+7

**Example 1:**

**Input**: N = 6, arr[] = {2, 3, 5, 6, 8, 10}

sum = 10

**Output:** 3

**Explanation**: {2, 3, 5}, {2, 8}, {10}

**Example 2:**

**Input**: N = 5, arr[] = {1, 2, 3, 4, 5}

sum = 10

**Output:** 3

**Explanation**: {1, 2, 3, 4}, {1, 4, 5},

{2, 3, 5}

**Your Task:**  
You don't need to read input or print anything. Complete the function **perfectSum()**which takes **N,**array**arr[]** and **sum**as input parameters and returns an integer value  
  
**Expected Time Complexity:** O(**N\*sum**)  
**Expected Auxiliary Space:** O(**N\*sum**)  
  
**Constraints:**  
1 ≤ **N\*sum** ≤ 106

class Solution{

    public:

    int mod=1e9+7;

    int Subsets(int i, int arr[], int sum, vector<vector<int>> &dp) {

        if (sum==0) return 1;

        if (i<0) return 0;

        if (dp[i][sum]!=-1) return dp[i][sum];

        if (arr[i]<=sum) {

            return dp[i][sum]=(Subsets(i-1, arr, sum-arr[i], dp)%mod+Subsets(i-1, arr, sum, dp)%mod)%mod;

        }

        else return dp[i][sum]=Subsets(i-1, arr, sum, dp);

    }

    int perfectSum(int arr[], int n, int sum) {

        // Your code goes here

        vector<vector<int>> dp(n+1, vector<int> (sum+1, 0));

        for (int i=0; i<n+1; i++) dp[i][0]=1;

        for (int i=1; i<n+1; i++) {

            for (int j=1; j<sum+1; j++) {

                if (arr[i-1]<=j) dp[i][j]=(dp[i-1][j-arr[i-1]]%mod+dp[i-1][j]%mod)%mod;

                else dp[i][j]=dp[i-1][j];

            }

        }

        return dp[n][sum];

    }

};