8) PARTITION EQUAL SUBSET SUM:

METHOD:
LINK OF THE EXPLANATION:
Partition equal subset sum | Equal sum partition | Dynamic Programming | Leetcode #416

1) The sum is very simple.
2) This is almost the same as the knapsack sum.Here we have the option of selecting the element to attain the required sum.
3) We simply neglect the elements which have the value greater than the required sum.
4) If the element is smaller than the required sum then we have the choice of selecting that element.
5) So we first call the function in which we select that element.If the function returns true then we don't call the function next for not selecting that element.
6) Otherwise we call the function for the choice of not selecting that element.

CODE OF THE PROGRAM:

```
class Solution{
public:
   bool fun(int arr[],int n,int index,int subsetsum,int requiredsum){
      if(subsetsum==requiredsum){
         return true;
      }
      else if(index==n){
         return false;
      }
      else{
         if(arr[index]+subsetsum<=requiredsum){
            bool status=fun(arr,n,index+1,subsetsum+arr[index],requiredsum);
            if(status==true){
               return true;
            }
            else{
               return fun(arr,n,index+1,subsetsum,requiredsum);
            }
         }
         else{
            return fun(arr,n,index+1,subsetsum,requiredsum);
         }
      }
   }

   int equalPartition(int n, int arr[]){
      int sum=0;
      for(int i=0;i<n;i++){
         sum=sum+arr[i];
      }
      if(sum%2!=0){
```

```
            return false;
        }
        else{
            return fun(arr,n,0,0,sum/2);
        }
    }
};
```