Count the total number of subsequnces

```
#include<bits/stdc++.h>
using namespace std;

int printSub(int index, int sum, int target, int *arr, int n){

    if( index == n){
        // condition satisfied
        if( sum  == target ){
            return 1;
        }
        // condition not satisfied
        else{
            return 0;
        }
    }

    // pick
    sum = sum + arr[index];
    int l = printSub( index + 1, sum, target, arr , n);


    sum = sum - arr[index];

    // Not pick
    int r = printSub( index + 1, sum, target, arr , n);

    return l + r;
}

int main(){
    int arr[3] = {1, 2, 1};
    int n = 3;
    int target = 2  ;
    cout << printSub(0, 0, target, arr, n);

    return 0;
}
```

Here, we magnify the count with recursion and good coding practice of not involving extra global variables, for each call it returns the sum of count returned by the left and right part. For the base case, we have return count 1, if the sum is achieved and count as 0, if the base case is not achieved.