

# Print the subsequences whose sum is k

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

void printF(int index, vector<int> &ans, int sum, int target, int arr[], int size){

    // base case :
    if( index == size){
        if( sum == target){
            for( auto i : ans){
                cout << i << " ";
            } cout << endl;
        }
        return;
    }

    // Picking
    ans.push_back( arr[index]);
    sum = sum + arr[index];
    printF( index + 1, ans, sum, target, arr, size);

    // Not Picking
    ans.pop_back();
    sum = sum - arr[index];
    printF( index + 1, ans, sum, target, arr, size);

    return;
}

int main(){
    int arr[3] = {1, 2, 1};
    int size = 3;
    int target = 2;
    vector<int> ans;
    printF( 0, ans, 0, target, arr, size );
}
```

- you are given one array as { 1, 2, 1 }
- you have the target sum as 2
- you have to find the subsequences that can make this sum
- 1 1 ( Pick - Not Pick - Pick)
- 2 ( Not pick, Pick, Not pick )

✓  
X

