Print the subsequences whose sum is k - you are given one array as { 1, 2, 1 } #include<bits/stdc++.h> - you have the target sum as 2 using namespace std; - you have to find the subsequences that can make this sum void printF(int index, vector<int> &ans, int sum, int target, int arr[], int size){ - 11 (Pick - Not Pick - Pick) - 2 (Not pick, Pick, Not pick) // base case : if(index == size){ if(sum == target){ for(auto i : ans){ cout << i << " "; } cout << endl;</pre> 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); f(3, 2, 4) f(2, 2, 3) 111 f(1, 1) 1--1 if () \ if () X here index has reached 3, now in this decision would be made if () X that if sum is 2 which is the target, then print the data structure we are forming f(3, {1,2,1},4) else if it is not 2 then simply return, yes just return do nothing. f(2, {1,2},3) if () X upper thing is returned now we will do the not picking upper thing is returned part, data structure is containing now we will do the not picking 1 which is element at index 2, part, data structure is containing so just remove the element from the 2 which is element at index 1, data structure and remove its impact so just remove the element from the on the sum data structure and remove its impact on the sum f(3, {1,2},3) f(2, {1, }, 1) if () \ return here index has reached 3, now in this decision would be made return that if sum is 2 which is the target, then print the data structure we are forming else if it is not 2 then simply return, yes just return do nothing. if () X return if () \ if () X here index has reached 3, now in this decision would be made that if sum is 2 which is the target, then print the data structure we are forming f(3, {1,1},2) 5 else if it is not 2 then simply return, yes just return do nothing. if () **√** upper thing is returned now we will do the not picking print 1, 1 part, data structure is containing 1 which is element at index 2, so just remove the element from the data structure and remove its impact on the sum f(0, f(3, {1}, 1) 1--1 if () X, if () \ return here index has reached 3, now in this decision would be made f(1, {1}, 1) that if sum is 2 which is the target, then print the data structure we are forming upper thing is returned else if it is not 2 then simply return, yes just return do nothing. now we will do the not picking if () X f(0,0,0) part, data structure is containing 1 which is element at index 0, so just remove the element from the data structure and remove its impact on the sum f(1, { }, 0) return f(2, 2, 2) if () X here index has reached 3, now in this decision would be made if () X that if sum is 2 which is the target, then print the data structure we are forming f(3, { 2, 1}, 3) else if it is not 2 then simply return, yes just return do nothing. f(2, {2}, 2) if () X upper thing is returned now we will do the not picking upper thing is returned part, data structure is containing return now we will do the not picking 1 which is element at index 2, part, data structure is containing so just remove the element from the 2 which is element at index 1, data structure and remove its impact so just remove the element from the on the sum data structure and remove its impact on the sum f(3, 2, 2) f(3, {2}, 2 f(2,{},) if () **/** return here index has reached 3, now in this decision would be made return that if sum is 2 which is the target, then print the data structure we are forming else if it is not 2 then simply return, yes just return do nothing. if () \ return f(3, 1, 1) if () X here index has reached 3, now in this decision would be made that if sum is 2 which is the target, then print the data structure we are forming f(3, {1}, 1) else if it is not 2 then simply return, yes just return do nothing. upper thing is returned if () X now we will do the not picking part, data structure is containing return 1 which is element at index 2, so just remove the element from the data structure and remove its impact on the sum f(3,{},0) if () \ • return here index has reached 3, now in this decision would be made

that if sum is 2 which is the target, then print the data structure we are forming

else if it is not 2 then simply return, yes just return do nothing.

if () X

return