Print only one subsequence whose sum is k out of all subsequences available

```
#include<bits/stdc++.h>
using namespace std;
bool 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;</pre>
            return true;
        return false;
    ans.push_back(arr[index]);
    sum = sum + arr[index];
    if(printF( index + 1, ans, sum, target, arr, size)){
        return true;
    ans.pop_back();
    sum = sum - arr[index];
    if(printF( index + 1, ans, sum, target, arr, size)){
        return true;
    return false;
int main(){
    int arr[3] = { 1,2, 1};
    int size = 3;
    int target = 2;
    vector<int> ans;
    printF(0, ans, 0, target, arr, size );
    return 0;
```

Just print the data structure when the sum is achieved and then return true in the base case.

Now in the main function you just need to return true and false, true has printing property associated with it.

What is the essence?

X

Essence is that we have made this printSubsequence function as the boolean function and the logic is that if the sum is achieved we return true, and along with this true we print the data structure, and this true just acts as the return medium

We could also use void ans do simply return?

Yes we could do that but we chose bool as we also used if to segregate and move to the next include or not include and inside if we need true or false to execute the statement so we made simple logic that if () means if expression inside it is true, then just return it recursively to end the function and print the subsequence and if (false) i.e. sum is not achieved move to next condition, which is checking of sum under pick and not pick.

if () $\sqrt{\text{sum is 2}}$

return true

print 1, 1

f(3, 2 , 4) if () x (index is not 3) if () \checkmark (index is 3) if () x (index is not 3) here index has reached 3, now in this decision would be made f(3, {1,2,1},4 that if sum is 2 which is the target, then print the false f(2, {1,2},3) data structure we are forming else if it is not 2, return false. false So the upper thing is false if () X sum is not 2 and then if(false) is there this turns out to be false, so the code inside this hence if (false) is there return false is not executed and we and statement inside the if move to next if which is not executed and we move to is exclusion of index 2 next if which is exclusion of element, 1. element at index 1 which is 2. > f(3, 2, 3) f(2, {1, }, 1) true f(3, {1,2},3) Now this returns true, this is also false so we know that if (true) if () $\sqrt{\text{(index is 3)}}$ and hence if (false) is there so statement inside this if is executed which is to so statement inside this here index has reached 3, now in this decision would be made is also not executed. return true directly and we not that if sum is 2 which is the target, then print the need to move to the next if data structure we are forming else if it is not 2, return false. if () X return false statement and this will if () x sum is not 2 f(0,0,0) directly return true. return false true if () x (index is not if () $\sqrt{\text{(index is 3)}}$ here index has reached 3, now in this decision would be made f(3, {1,1},2) that if sum is 2 which is the target, then print the true data structure we are forming else if it is not 2, return false.

Now this returns true,

directly return true.

so we know that if (true)

is there so statement inside this if

is executed which is to return true

the next if statement and this will

directly and we not need to move to