```
//
//  main.cpp
//  C++ Terminal
//
//  Created by Jiaqi Li on 2019/12/29.
//  Copyright © creation date – 2020 Jiaqi Li. All rights reserved.
//
//  Combinational sum calculator
//  Input an array of numbers and a sum number, end array with ","
//  Output all combinations that sums up to sum in ascending order

#include <iostream>
#include <algorithm>
#include <array>
#include <cmath>
#include <vector>
#include <numeric>
using namespace std;
//combinational sum, vector, malloc, EOF, sorting, recurrence, functions

///////////////functional functions to sum up elements of an array
int vectorsum(vector<int> vec){
    
    int sum = accumulate(vec.begin(), vec.end(),0);
    
    return sum;
};

//stuff an array with designated element for a given times
vector<int> stuffvector(int times, vector<int> vec, int element){
    
    for (int i = 1; i < times+1; i++){
        vec.push_back(element);
    };
    
    return vec;
};

//print out the array
void printvector(vector<int> vecarr){
    cout << "[";
    for(int i=0; i < vecarr.size(); i++){
        if (i != vecarr.size()-1){
            cout << vecarr.at(i) << ", ";
        }else{
            cout << vecarr.at(i);
        }
    };
    cout << "]" << endl;
};

/////////////functions that create the answer array (start reading at the
 second block then go to the first)
```

```
//the main maths here is to 'stuff the answer array' with a number of the
 same element from the input array, this number is chosen such that
 number*element <= sum

//the difference between the createanswer function and the ampcreateanswer
 function is that ampcreateanswer removes the first element of the input
 vector with each iteration to consider the possiblility of adding the
 subsequent elements into the answer array
vector<int> ampcreateanswer(vector<int> vecarr, int counter, int tempsum,
 vector<int> outputarr, const int sum){
    vector<int> original = outputarr;

    vector<int> tempvecarr = vecarr;
    tempvecarr.erase(tempvecarr.begin(), tempvecarr.begin()+1);

    for (counter; counter < vecarr.size(); counter++){

        if (vectorsum(outputarr) != sum && vectorsum(outputarr) < sum){
            int arraysum = 0;
            int target = 0;

            target = vecarr[counter];
            int multiplier = (sum-vectorsum(outputarr))/target;
            if (multiplier == 0){
                multiplier = 1;
            };

            if (sum - vectorsum(outputarr) == multiplier * target){
                outputarr = stuffvector(multiplier, outputarr, target);
            }else{
                for(int counter2 = multiplier; counter2 > 0; counter2--){
                    outputarr = stuffvector(counter2, outputarr, target);
                    arraysum = vectorsum(outputarr);
                    if (vectorsum(outputarr) < sum) {
                    outputarr = ampcreateanswer(tempvecarr, counter+1,
                     tempsum-arraysum, outputarr, sum);
                    };
                };
            }
        };
        if (vectorsum(outputarr)!=sum){
            outputarr = original;
        }
    };
    return outputarr;
};

//this is the main function, it stuffs the output array with multiple
 instances of the selected element and ampcreateanswer takes similar
 actions to finish the output array, if the output/answer array is
 satisfactory the function will print it out, clear the array and move to
 the second possible answer combination
void createanswer(vector<int> vecarr, int counter, int tempsum, vector<int>
 outputarr, int &answerNR, const int sum){
    int tempanswerNR = 0;
    for (counter; counter < vecarr.size(); counter++){
        int arraysum = 0;
        int target = 0;

        target = vecarr[counter];
        int multiplier = sum/target;

        for(int counter2 = multiplier; counter2 > 0; counter2--){
            outputarr = stuffvector(counter2, outputarr, target);
            arraysum = vectorsum(outputarr);
            if (vectorsum(outputarr) < sum) {
                outputarr = ampcreateanswer(vecarr, counter+1,
                 tempsum-arraysum, outputarr, sum);
            };
            if (vectorsum(outputarr) == sum){
                printvector(outputarr);
                outputarr.clear();
                tempanswerNR++;
                answerNR = tempanswerNR;
            } else {
                outputarr.clear();
            };
        };
    };
};


int main() {
    //input array via vector and find sum
    int element = 0;
    int sum = 0;
    vector<int> vecarr;
    while(cin>>element) {
        vecarr.push_back(element); //sentinel value = ,
    };
    sum = vecarr[vecarr.size()-1];
    cout << "The sum we are looking for is: " << sum << "." << endl;
    vecarr.pop_back();
    sort(vecarr.begin(), vecarr.end());
    cout << "Input Vector is: ";
    printvector(vecarr);

    //do combinational sum (see functions)
    int counter = 0;
    int answerNR=0;
    vector<int> outputarr;
    createanswer(vecarr, counter, sum, outputarr, answerNR, sum);
    if (answerNR == 0){
        cout << "Empty" << "\n";
    }
    return 0;

}
```