#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

void displaySet(vector<int>subset){

for(int j=0;j<subset.size();j++)

cout<<subset[j]<<" ";

cout<<"\n";

}

void subsetSum(vector <int> weight, vector <int> subset,int m,int i,int cur,int max,int &count){

int n = weight.size();

if(cur == m){

//subset found

count++;

displaySet(subset);

return;

}

if(i>=n || cur + max < m || cur + weight[i] > m )

return;

// with ith element

subset.push\_back(weight[i]);

subsetSum(weight,subset,m,i+1,cur + weight[i],max-weight[i],count);

// without the ith element

subset.pop\_back();

subsetSum(weight,subset,m,i+1,cur,max-weight[i],count);

}

int main(){

int n,sum,count=0;

vector<int>subset;

vector <int> weight;

cout<<"enter the number of elements: ";

cin>>n;

weight.resize(n);

cout<<"enter the weights: ";

for(int i=0;i<n;i++)

cin>>weight[i];

cout<<"enter the required sum: ";

cin>>sum;

// the weights are sorted in ascending order

sort(weight.begin(),weight.end());

int max=0;

for(int j=0;j<n;j++)

max += weight[j];

cout<<"the possible subsets are: \n";

subsetSum(weight,subset,sum,0,0,max,count);

if(count)

cout<<"the number of subsets are: "<<count;

else

cout<<"no subsets possible";

return 0;

}

OUTPUT:

enter the number of elements: 5

enter the weights: 2 4 6 8 10

enter the required sum: 20

the possible subsets are:

2 4 6 8

2 8 10

4 6 10

the number of subsets are: 3

enter the number of elements: 10

enter the weights: 1 2 3 4 5 6 7 8 9 10

enter the required sum: 11

the possible subsets are:

1 2 3 5

1 2 8

1 3 7

1 4 6

1 10

2 3 6

2 4 5

2 9

3 8

4 7

5 6

the number of subsets are: 11

enter the number of elements: 3

enter the weights: 1 1 1

enter the required sum: 4

the possible subsets are:

no subsets possible