**PROBLEM STATEMENT**:

Given a set of candidate numbers (C) and a target number (T), find all unique combinations in C where the candidate numbers sums to T.

The same repeated number may be chosen from C unlimited number of times.

**Example**,  
Given candidate set 2,3,6,7 and target 7,  
A solution set is:

[2, 2, 3]

[7]

**CODE:**

void util(vector<vector<int>>&ans,vector<int>part,vector<int>ar,int sum,int idx)

{

if(sum==0)

{

ans.push\_back(part);

return;

}

for(int i=idx;i<ar.size();i++)

{

if(ar[i]>sum)

break;

part.push\_back(ar[i]);

util(ans,part,ar,sum-ar[i],i);

part.pop\_back();

}

}

vector<vector<int> > Solution::combinationSum(vector<int> &A, int B) {

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

vector<vector<int>>ans;

vector<int>part;

util(ans,part,A,B,0);

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

ans.erase(unique(ans.begin(),ans.end()),ans.end());

return ans;

}