public class Solution {

public ArrayList<ArrayList<Integer>> combinationSum(int[] candidates, int target) {

ArrayList<ArrayList<Integer>> res = new ArrayList<ArrayList<Integer>>();

ArrayList<Integer> tmp = new ArrayList<Integer>();

Arrays.sort(candidates);

dfs(candidates,0,target,tmp,res);

return res;

}

private void dfs(int[] candidates, int start, int target, ArrayList<Integer> tmp,

ArrayList<ArrayList<Integer>> res)

{

if(target<0)

return;

if(target==0)

{

res.add(new ArrayList<Integer>(tmp));

return;

}

for(int i=start;i<candidates.length;i++)

{

if(i>0 && candidates[i]==candidates[i-1])

continue;

tmp.add(candidates[i]);

dfs(candidates,i,target-candidates[i],tmp,res);

tmp.remove(tmp.size()-1);

}

}

}