// C++ program to find all subsets of given set. Any

// repeated subset is considered only once in the output

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void findSubsets(int ind, vector<int>& nums, vector<int>& ds, vector<vector<int>>& ansList) {

ansList.push\_back(ds);

for (int i = ind; i < nums.size(); i++) {

if (i != ind && nums[i] == nums[i - 1])

continue;

ds.push\_back(nums[i]);

findSubsets(i + 1, nums, ds, ansList);

ds.pop\_back();

}

}

vector<vector<int>> AllSubsets(int arr[], int n) {

vector<int> nums(arr, arr + n);

vector<int> ds;

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

vector<vector<int>> ansList;

findSubsets(0, nums, ds, ansList);

return ansList;

}

int main() {

int set[] = { 10, 12, 12 };

vector<vector<int>> subsets = AllSubsets(set, 3);

for (auto subset : subsets) {

cout << "[";

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

cout << subset[i];

if (i < subset.size() - 1) {

cout << ", ";

}

}

cout << "], ";

}

return 0;

}