#include <iostream>

#include <vector>

using namespace std;

void calcSubset(vector<int>& A, vector<vector<int> >& res,

vector<int>& subset, int index)

{

// Add the current subset to the result list

res.push\_back(subset);

// Generate subsets by recursively including and

// excluding elements

for (int i = index; i < A.size(); i++) {

// Include the current element in the subset

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

// Recursively generate subsets with the current

// element included

calcSubset(A, res, subset, i + 1);

// Exclude the current element from the subset

// (backtracking)

subset.pop\_back();

}

}

vector<vector<int> > subsets(vector<int>& A)

{

vector<int> subset;

vector<vector<int> > res;

int index = 0;

calcSubset(A, res, subset, index);

return res;

}

// Driver code

int main()

{

vector<int> array = { 1, 2, 3 };

vector<vector<int> > res = subsets(array);

// Print the generated subsets

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

for (int j = 0; j < res[i].size(); j++)

cout << res[i][j] << " ";

cout << endl;

}

return 0;

}