#include <iostream>

#include <vector>

using namespace std;

void findSubsets(int arr[], int n, int target, vector<int> &path, int sum, int index)

{

if (sum == target)

{

cout << "Selected -> ";

for (int val : path)

{

cout << val << " ";

}

cout << "\tSum: " << target << " \tNot selected -> ";

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

{

bool selected = false;

for (int val : path)

{

if (val == arr[i])

{

selected = true;

break;

}

}

if (!selected)

{

cout << arr[i] << " ";

}

}

cout << endl; // visaru nko

return;

}

if (index >= n || sum > target)

{

return; // this loop also

}

// Include current element

path.push\_back(arr[index]);

findSubsets(arr, n, target, path, sum + arr[index], index + 1);

// Exclude current element

path.pop\_back();

findSubsets(arr, n, target, path, sum, index + 1);

}

int main()

{

int n;

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

cin >> n;

int arr[n];

cout << "Enter the elements: ";

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

{

cin >> arr[i];

}

int target;

cout << "Enter the target sum: ";

cin >> target;

vector<int> path;

findSubsets(arr, n, target, path, 0, 0);

return 0;

}