**#include <iostream>**

**using namespace std;**

**void displaySubset(int subSet[], int size) {**

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

**cout << subSet[i] << " ";**

**}**

**cout << endl;**

**}**

**void subsetSum(int set[], int subSet[], int n, int subSize, int total, int nodeCount ,int sum) {**

**if( total == sum) {**

**displaySubset(subSet, subSize); //print the subset**

**subsetSum(set,subSet,n,subSize-1,total-set[nodeCount],nodeCount+1,sum); //for other subsets**

**return;**

**}else {**

**for( int i = nodeCount; i < n; i++ ) { //find node along breadth**

**subSet[subSize] = set[i];**

**subsetSum(set,subSet,n,subSize+1,total+set[i],i+1,sum); //do for next node in depth**

**}**

**}**

**}**

**void findSubset(int set[], int size, int sum) {**

**int \*subSet = new int[size]; //create subset array to pass parameter of subsetSum**

**subsetSum(set, subSet, size, 0, 0, 0, sum);**

**delete[] subSet;**

**}**

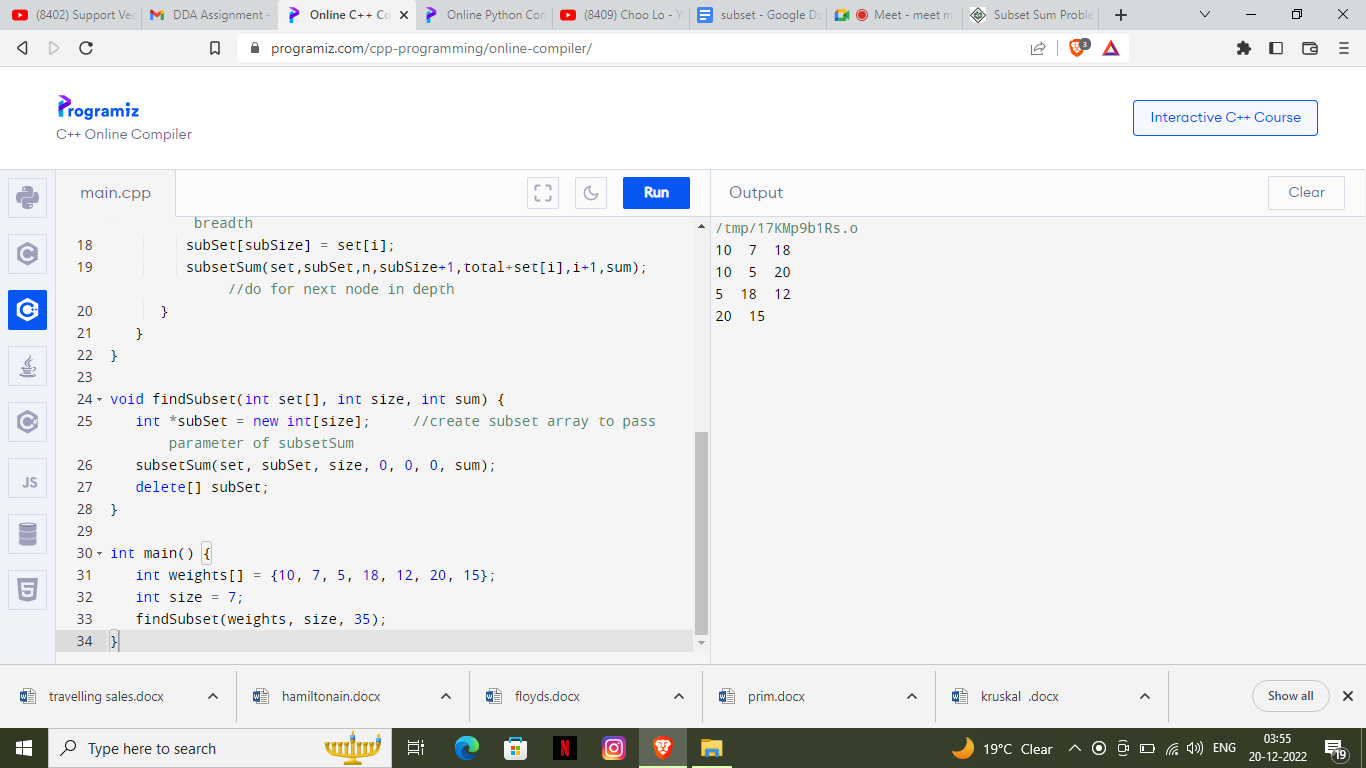
**int main() {**

**int weights[] = {10, 7, 5, 18, 12, 20, 15};**

**int size = 7;**

**findSubset(weights, size, 35);**

**}**

****