Sum of sets

**#include <stdio.h>**

**#include <stdlib.h>**

**Static int total\_nodes;**

**Void printValues(int A[], int size){**

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

**Printf(“%\*d”, 5, A[i]);**

**}**

**Printf(“\n”);**

**}**

**Void subset\_sum(int s[], int t[], int s\_size, int t\_size, int sum, int ite, int const target\_sum){**

**Total\_nodes++;**

**If (target\_sum == sum) {**

**printValues(t, t\_size);**

**subset\_sum(s, t, s\_size, t\_size – 1, sum – s[ite], ite + 1, target\_sum);**

**return;**

**}**

**Else {**

**For (int i = ite; i < s\_size; i++) {**

**T[t\_size] = s[i];**

**Subset\_sum(s, t, s\_size, t\_size + 1, sum + s[i], i + 1, target\_sum);**

**}**

**}**

**}**

**Void generateSubsets(int s[], int size, int target\_sum){**

**Int\* tuplet\_vector = (int\*)malloc(size \* sizeof(int));**

**Subset\_sum(s, tuplet\_vector, size, 0, 0, 0, target\_sum);**

**Free(tuplet\_vector);**

**}**

**Int main(){**

**Int set[] = { 5, 6, 12 , 54, 2 , 20 , 15 };**

**Int size = sizeof(set) / sizeof(set[0]);**

**Printf(“The set is “);**

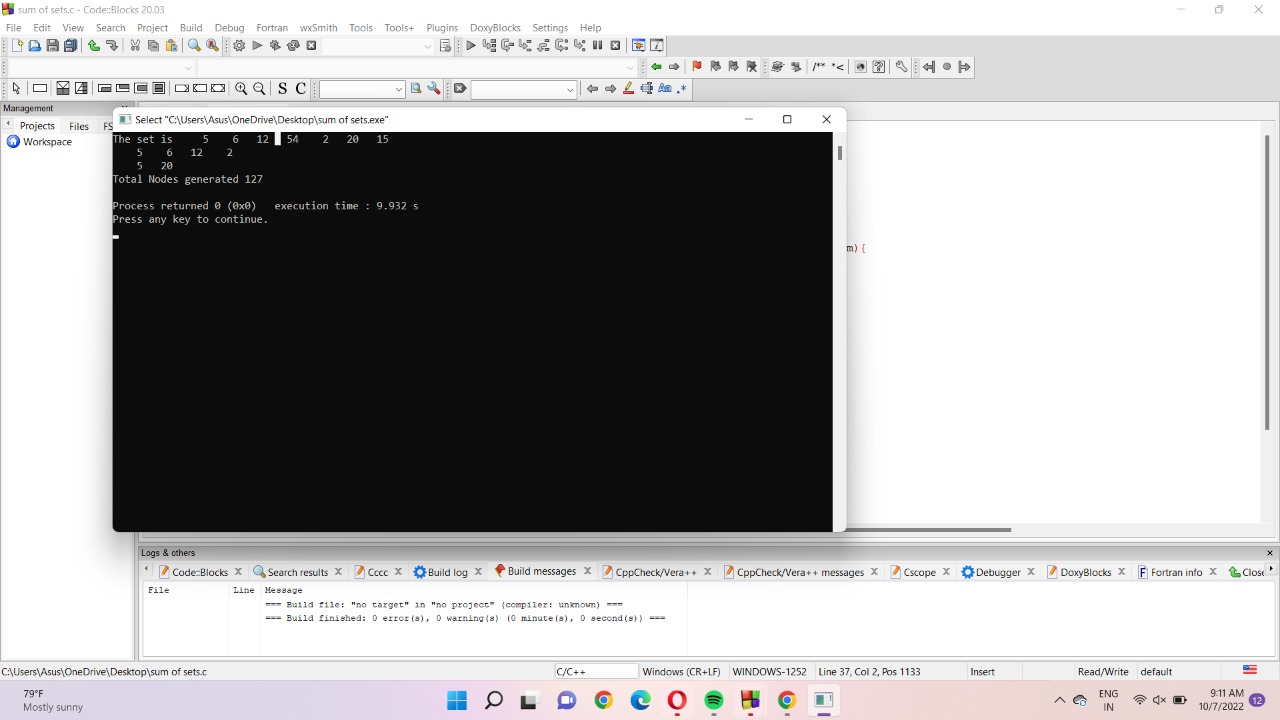
**printValues(set , size);**

**generateSubsets(set, size, 25);**

**printf(“Total Nodes generated %d\n”, total\_nodes);**

**return 0;**

**}**

****