***NAME : YASH GHULE***

***ROLL NO : 2193289***

***CLASS : CSE CORE 2***

***ASSIGNMENT – 3***

3. IMPLEMENT SUM OF SUBSET .

#include <bits/stdc++.h>

using namespace std;

#define ARRAYSIZE(a) (sizeof(a))/(sizeof(a[0]))

static int total\_nodes;

void printSubset(int A[], int size)

{

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

{

cout<<" "<< A[i];

}

cout<<"\n";

}

int comparator(const void \*pLhs, const void \*pRhs)

{

int \*lhs = (int \*)pLhs;

int \*rhs = (int \*)pRhs;

return \*lhs > \*rhs;

}

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 )

{

printSubset(t, t\_size);

if( ite + 1 < s\_size && sum - s[ite] + s[ite + 1] <= target\_sum )

{

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

}

return;

}

else

{

if( ite < s\_size && sum + s[ite] <= target\_sum )

{

for( int i = ite; i < s\_size; i++ )

{

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

if( sum + s[i] <= target\_sum )

{

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));

int total = 0;

qsort(s, size, sizeof(int), &comparator);

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

{

total += s[i];

}

if( s[0] <= target\_sum && total >= target\_sum )

{

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

}

free(tuplet\_vector);

}

int main()

{

int weights[] = {15, 22, 14, 26, 32, 9, 16, 8};

int target = 53;

int size = ARRAYSIZE(weights);

generateSubsets(weights, size, target);

cout << "Nodes generated " << total\_nodes;

return 0;

}

OUTPUT :

