NAME: RAMYA RAMESH

USN: 1BM19CS227

**PROGRAM 17: Implement “Sum of Subsets” using Backtracking. “Sum of Subsets” problem: Find a subset of a given set S = {s1,s2,……,sn} of n positive integers whose sum is equal to a given positive integer d**

CODE:

#include<stdio.h>

int s[10] , x[10],d ;

void sumofsub ( int , int , int ) ;

void main ()

{

int n , sum = 0 ;

int i ;

printf ( " \n Enter the size of the set : " ) ;

scanf ( "%d" , &n ) ;

printf ( " \n Enter the set in increasing order:\n" ) ;

for ( i = 1 ; i <= n ; i++ )

scanf ("%d", &s[i] ) ;

printf ( " \n Enter the value of d : \n " ) ;

scanf ( "%d" , &d ) ;

for ( i = 1 ; i <= n ; i++ )

sum = sum + s[i] ;

if ( sum < d || s[1] > d )

printf ( " \n No subset possible " ) ;

else

sumofsub ( 0 , 1 , sum ) ;

}

void sumofsub ( int m , int k , int r )

{

int i=1 ;

x[k] = 1 ;

if ( ( m + s[k] ) == d )

{

printf("Subset:");

for ( i = 1 ; i <= k ; i++ )

if ( x[i] == 1 )

printf ( "\t%d" , s[i] ) ;

printf ( "\n" ) ;

}

else

if ( m + s[k] + s[k+1] <= d )

sumofsub ( m + s[k] , k + 1 , r - s[k] ) ;

if ( ( m + r - s[k] >= d ) && ( m + s[k+1] <=d ) )

{

x[k] = 0;

sumofsub ( m , k + 1 , r - s[k] ) ;

}

}

OUTPUT:





