ANAGHA ACHARYA

1BM19CS224

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. For example, if S = {1,2,5,6,8} and d = 9 there are two solutions {1,2,6} and {1,8}. A suitable message is to be displayed if the given problem instance doesn’t have a solution.

#include<stdio.h>

int x[10],w[10],d,count=0;

void subset(int cs,int k,int r)

{

int i;

x[k]=1;

if((cs+w[k])==d)

{

printf("\nSubset %d: ",++count);

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

if(x[i]==1)

printf("%d ",w[i]);

}

else

if(cs+w[k]+w[k+1]<=d)

subset(cs+w[k],k+1,r-w[k]);

if(cs+r-w[k]>=d && cs+w[k]<=d)

{

x[k]=0;

subset(cs,k+1,r-w[k]);

}

}

void main()

{

int i,n,sum=0;

printf("Enter the number of elements: ");

scanf("%d",&n);

printf("\nEnter the elements in ascending order: ");

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

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

printf("\nEnter the sum(d): ");

scanf("%d",&d);

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

sum=sum+w[i];

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

{

printf("No subset possible");

}

else{

subset(0,1,sum);

}

}

OUTPUT

