8. Design and implement C/C++ Program to find a subset of a given set S = {sl , s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d.

Objectives: This Program enable students to :

- Learn Subset problem.
- Find SUM is equal to a given positive integer

Aim:

An instance of the Subset Sum problem Is a pair(S,t),where S={x1,x2,.....,xn} is a set of Positive integers and t (the target) is a positive integer. The decisionproblemasks for a subset of S whose sum is as large as possible, but not larger than t.

This problem is NP-complete.

This problem arises in practical applications. Similar to the knapsack problem we may have a truck that can carry at most t pounds and we have n different boxes to ship and the ith box weighs xi pounds. The naive approach of computing the sum of the elements of every subset of S and then selecting the best requires exponential time. Below we present an exponential time exact algorithm.

```
#include<stdio.h>
void subset(int,int,int);
int x[10],w[10],d,count=0;
void main()
{
int i,n,sum=0;

printf("Enter the no. of elements: ");
scanf("%d",&n);
printf("\nEnter the elements in ascending order:\n");
for(i=0;i<n;i++)
scanf("%d",&w[i]);
printf("\nEnter the sum: ");
scanf("%d",&d);
for(i=0;i<n;i++)
sum=sum+w[i];
if(sum<d)
{
printf("No solution\n");
return;
}
subset(0,0,sum);
if(count==0)
{
printf("No solution\n");
return;
}
}
void subset(int cs,int k,int r)
{
int i; x[k]=1;
if(cs+w[k]==d)
{
printf("\n\nSubset %d\n",++count);
for(i=0;i<=k;i++)
```

```
if(x[i]==1)
 printf("%d\t",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]);
}
}
```