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. 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.

Program:

```
#include<stdio.h>

#include<conio.h>

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

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

{

int i;

x[k]=1;

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

{

printf("\nSubset solution = %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+1])<=d)

{

x[k]=0;

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

}

    }

void  main()

   {

int sum=0,i,n;

printf("Enter the number of elements\n");
```

```
scanf("%d", &n);

printf("Enter the elements in ascending order\n");

for(i=0;i<n;i++)

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


printf("Enter the required sum\n");

scanf("%d", &d);

for(i=0;i<n;i++)

sum+=w[i];

if(sum<d)

{

printf("No solution exists\n");

return;

}

printf("The solution is\n");

count=0;

subset(0,0,sum);

}
```

OutputScreenshot: