11. Design and implement in Java 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. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.

```
import java.util.Scanner;
public class SubsetDemo
{
   static int x[] = new int[20];
   static int s[] = new int[20];
   static int d,flag=0;
   public static void main(String [] args)
   {
    int i,n,sum=0;

    Scanner in = new Scanner(System.in);
    System.out.println("Enter the  no of elements");
    n = in.nextInt();

    System.out.println("Enter the elements");
    for(i=1;i<=n;i++)
    s[i] = in.nextInt();

     System.out.println("Enter the value of d:");
    d = in.nextInt();

    for(i=1;i<=n;i++)
    {
      sum = sum + s[i];
    }
    // Check for the problem instance not having the solution
    if(sum<d || s[1]>d)
    {
     System.out.println("The given problem instance does not have a solution");
     System.exit(0);

    }
    else
    {
       System.out.println("Subsets are:");
       SumofSub(0,1,sum);
    }

    if(flag ==0)
    {
      System.out.println("No subset possible");
```

```
      System.exit(0);
    }
   }
   // Method to Calculate the Subset Sum
   public static void SumofSub(int m,int k,int r)
   {
    int i;
    x[k] = 1;
    if(m+s[k] == d)
    {
     flag =1;

     System.out.print("{");
     for(i=1;i<=k;i++)
     {
      if(x[i] == 1)
      System.out.print(s[i] + " ");
     }
     System.out.print("}\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]);
    }
   }
}
```