Experiment 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.*;
public class subset {
	static int d,c=0;
	static int w[]= new int[10];
	static int x[]= new int[10];
	public static void main(String[] args) {
		int sum=0,i;
		Scanner sc= new Scanner(System.in);
		System.out.println("enter the numberof weights");
		int n= sc.nextInt();
		System.out.println("enter the weights in the ascending order");
		for(i=1;i<=n;i++)
			w[i]= sc.nextInt();
		System.out.println("enter the value of Sum");
		d= sc.nextInt();
		for(i=1;i<=n;i++)
			sum=sum+w[i];
		if(sum<d||w[1]>d)
		{
			System.out.println("Subset not possible as total weight is
less than given sum");
			System.exit(0);
		}
		sum_subset(0,1,sum);
		if(c==0)
			System.out.println("subset not possible");
	}
	static void sum_subset(int m ,int k, int r)
	{
		int i;
		x[k]=1;
		if((m+w[k]==d))
				{
			c++;
			System.out.println("Subset");
			for(i=1;i<=k;i++)
			{
				if(x[i]==1)
					System.out.println("\t" + w[i]);
			}
				}
		else
		{
			if(m+w[k]+w[k+1]<=d)
				sum_subset(m+w[k],k+1,r-w[k]);
			if((m+r-w[k]>=d) && m+w[k+1]<=d)
			{
				x[k]=0;
				sum_subset(m,k+1,r-w[k]);
			}
		}

			}
	}
```

//output:

```
enter the number of weights

4
enter the weights in the ascending order
1
3
5
7
enter the value of Sum
11
Subset
       1
       3
       7


enter the number of weights
5
enter the weights in the ascending order
1
2
5
6
8
enter the value of Sum
4
subset not possible
```