Subject :	Software :
	Hardware :
Branch : / Semester :	Page No. / Prog No. 10(i)

PROBLEM STATEMENT

Design and impliment c++/Java Program to find a subset of a given set $S = \{s1, s2, \ldots, sn\}$ 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.

ALGORITHM & CODE:

Program:

```
import Java.util.ArrayList;
import Java.util.List;
import Java.util.Scanner;
Public Class SubsetSumWithUserInput
{ Public Static void main (String[] args)
  { Scanner scanner = new Scanner (System.in);
    System.out.Println ("Enter the number of elements in
the set: ");
    int n = scanner.nextInt();
    int[] s = new int[n];
    System.out.Println ("Enter the elements of the set (one by
one): ");
    for (int i = 0; i<n; i++)
    { s[i] = scanner.nextInt ();
    }
    System.out.Println ("Enter the target Sum (Positive integer):");
    int d = scanner.nextInt ();
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :	Signature of Faculty Date :	Signature of Student Date :

Subject :	Software :
	Hardware :
Branch : / Semester :	Page No. / Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
System.out.Println ("\n Input set :");
PrintSet (S);
System.out. Println ("Target sum: "+ d);
List <List <Integer>> Solutions = findSubsetsWithSum
                                  (S, d );
if (Solutions. isEmpty())
{ System.out.Println ("\n No Subset found that sums
to "+d );
}
else {
    System.out.Println ("\n found "+ Solutions. size ()+
"Solution(s): ");
    {System.out. Println (solution);
    }
}
Scanner. close ();
}
Public static List< List <Integer >> findSubsetsWithSum
(int[] S , int d)
{
List <List <Integer >> Solutions = new ArrayList <>();
backtrack (S, d, 0, new ArrayList <> (), Solutions, 0);
return Solutions;
}
```

INPUT GIVEN	
OUTPUT OBTAINED	
REMARKS	
GRADE :	Signature of Faculty Date : / Signature of Student Date :

Subject :		Software :	
		Hardware :	
Branch :	Semester :	Page No.	Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
Private static void backtrack (int [] S, int
target, int Start, List <Integer> Current, List<
List <Integer>> Solutions , int currentSum)
  {if (currentSum == = target)
    { Solutions. add (new ArrayList <> (Current));
      return;
    }
    for (int i = Start ; i <S.length ; i++)
    { if (currentSum + S[i] > target)
      {continue;
      }
      Current. add (S[i]);
      backtrack (S, target, i +1, Current, Solutions,
currentSum + S[i] );
          Current. remove (Current. size () - 1 );
    }
  }
Private Static void PrintSet (int [] set )
{System.out. Print ("{");
  for (int i = 0 ; i < set. length ; i++)
  { System. out. Print (set[i] );
    if (i< set. length - 1)
```

INPUT GIVEN	
OUTPUT OBTAINED	
REMARKS	
GRADE :	Signature of Faculty Date : / Signature of Student Date :

Subject :	Software :
	Hardware :
Branch : / Semester :	Page No. / Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
        {
            System.out.Print(", ");
        }
    }
    System.out.Println("}");
}
```

OUTPUT

```
Enter the number of elements in the set: 5
Enter the elements of the set (one by one): 1 2 5 6 8
Enter the target sum (positive integer): 9

input set :
(1, 2, 5, 6, 8)
Target Sum : 9
Found 2 solution(s):
[1, 2, 6]
[1, 8]
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty
Date :

Signature of Student
Date :