Experiment 2.2

Aim: *Develop a program and analyze complexity to implement subset-sum problem using Dynamic Programming.*

Objectives: *Objective is to implement subset-sum problem using Dynamic programming.*

Input/Apparatus Used: *VS CODE*

Procedure/Algorithm:

- *So we will create a 2D array of size (arr.size() + 1) * (target + 1) of type boolean. The state DP[i][j] will be true if there exists a subset of elements from A[0….i] with sum value = 'j'. The approach for the problem is:*

  if

  (A[i-1] > j) DP[i][j] = DP[i-1][j]

  else

  DP[i][j] = DP[i-1][j] OR DP[i-1][j-A[i-1]]
- *This means that if current element has value greater than 'current sum value' we will copy the answer for previous cases*
- *And if the current sum value is greater than the 'ith' element we will see if any of previous states have already experienced the sum='j' OR any previous states experienced a value 'j – A[i]' which will solve our purpose.*

Code:

```
import java.io.*;
class DAAexp6 {
   static boolean isSubsetSum(int set[], int n, int sum)
   {
      boolean subset[][] = new boolean[sum + 1][n + 1];
      for (int i = 0; i <= n; i++)
         subset[0][i] = true;
      for (int i = 1; i <= sum; i++)
         subset[i][0] = false;
```

```
        for (int i = 1; i <= sum; i++) {
            for (int j = 1; j <= n; j++) {
                subset[i][j] = subset[i][j - 1];
                if (i >= set[j - 1])
                    subset[i][j]
                        = subset[i][j]
                          || subset[i - set[j - 1]][j - 1];
            }
        }
        return subset[sum][n];
    }
    public static void main(String args[])
    {
        int set[] = { 3, 34, 4, 12, 5, 2 };
        int sum = 9;
        int n = set.length;
        if (isSubsetSum(set, n, sum) == true)
            System.out.println("Found a subset"
                               + " with given sum");
        else
            System.out.println("No subset with"
                               + " given sum");
    }
}
```

Observations/Outcome :

```
3250f27c81253090c720\redhat.java\jdt_ws\DAA_30d42fd\bin' 'DAAexp6'
Found a subset with given sum
PS E:\NOTES\Sem 5\DAA>
```

Time Complexity:

- *Time Complexity: O(n * targetSum),*
- *Space Complexity: O(n * targetSum)*