// w = [0] + input list , i = no. of elements , M = Max. Sum

w = [0, 4, 8, 9, 10, 12, 15] ; i = 6 ; M = 30 ⇐ e.g.,

Algorithm SubsetSumM (i, M)

```
if (i == 0):  // no elements
    return 0, []
elif (M - wi < 0): // weight of ith element > M
    return SubsetSumM (i-1, M)
elif (M - wi == 0): // weight of ith element = M
    return wi, [wi]
else:
    val1, lst1 = SubsetSumM (i-1, M) // not including ith element
    val2, lst2 = SubsetSumM (i-1, M - wi) // including ith element
    val2 += wi
    if (val2 > val1):
        return val2, lst2. + [wi]
    else:
        return val1, lst1
```

Recurrence Relation:

$$S(i, M) = \max \{ S(i-1, M), w_i + S(i-1, M-w_i) \}$$

k

$$S(0, M) = 0 \quad , \quad S(i, M) = -\infty \text{ for } M < 0$$