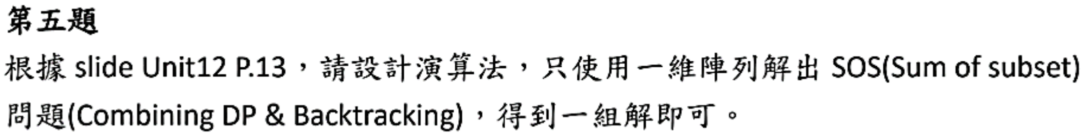

第五題

根據 slide Unit12 P.13，請設計演算法，只使用一維陣列解出 SOS(Sum of subset)問題(Combining DP & Backtracking)，得到一組解即可。

題目：Use 1 dimension array solve SOS problem

演算法：由於原本的演算法也只用的到上一層的 array 所以可以轉換成一維，陣列大小就是 M+1 然後從 M 跑到 1 避免重複放入 element

最後要找到子集合元素只需要判斷 dp[i-S[e]] 是否=True，若是 True 則把 S[e]放入 subset

Pseudocode：

```
S is a set of input numbers
dp = array of M+1 element of boolean

// init
All element of dp are False
dp[0] = True

//run algorithm
for e from 0 to (length of S )-1
    for i from M to 0
        if i < S[e]
            break
        if dp[i] or dp[i - S[e]]
            dp[i] = True
//get subset
if dp[M] == 1
    v = M
    elementindex = len(S)-1
    subset = []
    while(v != 0)
        if S[elementindex] <= v and dp[v-S[elementindex]] == True
            subset.append(S[elementindex])
            v = v-S[elementindex]
        elementindex -= 1
    print("Exist")
    print(subset)
else
    print("Not exist any subset that sum of subset equal M")
```