Given n numbers $x_1, x_2, \cdots, x_n$, consider the problem of computing $d[i, j] = x_i + x_{i+1} + \cdots + x_j$, for all i<=j. A naïve algorithm by computing each $d[i, j]$ independently will take $\Theta(n^3)$ time. Derive an efficient way to solve this problem in $O(n^2)$ time. 【98 年中央資工所】

Ans.

$d[i,j] = x_i + x_{i+1} + \cdots + x_j$, for all $i \le j$

```
for i ← 1 to n {
        d[i,i] = x_i ;
}
for len ← 2 to n {
        for i ← 1 to n-len+1 {
            j = i+len-1 ;
            d[i,j] = d[i,i] + d[i+1,j] ;
        }
}
```