0-1 Knapsack : DP algorithm

The dynamic programming algorithm is now (more or less) straightforward.

```
function 0-1-Knapsack(w, v, n, W)
  int K[n, W + 1];
  for(i = 1; i ≤ n; i++)   K[i, 0] = 0;
  for(j = 0; j ≤ W; j++)
     if (w[1] ≤ j) then K[1, j] = v[1];
     else K[1, j] = 0;
  for (i = 2; i ≤ n; i++)
      for (j = 1; j ≤ W; j++)
          if (j ≥ w[i] && K[i − 1, j − w[i]] + v[i] > K[i − 1, j])
             K[i, j] = K[i − 1, j − w[i]] + v[i];
          else
             K[i, j] = K[i − 1, j];
  return K[n, W];
```