Algorithm 1 Knapsack Algorithm - Dynamic Programming

```
1: procedure KNAPSACK(M)
2:     cost ← M + 1 array of 0's
3:     best ← M + 1 array of 0's
4:     for i from 1 to N do
5:         for k from size[i] to M do
6:             if val[i] + cost[k-size[i]] > cost[k] then
7:                 cost[k] = val[i] + cost[k-size[i]]
8:                 best[k] = i
9:     print(cost[M])
10:    for k from M to 0 step size[best[k]] do
11:        print(best[k])
```