EXAMPLE: **SOLVING KNAPSACK PROBLEM WITH DYNAMIC PROGRAMMING**

Selection of n=4 items, capacity of knapsack M=8

Item i	Value v_i	Weight w_i
1	15	1
2	10	5
3	9	3
4	5	4

f(0,g) = 0, f(k,0) = 0

Recursion formula:

$$f(k,g) = \begin{cases} f(k-1,g) & \text{if } w_k > g \\ \max\{v_k + f(k-1,g-w_k),\ f(k-1,g)\} & \text{if } w_k \leq g \text{ and } k>0 \end{cases}$$

Solution tabulated:

		Capacity remaining g=0	g=1	g=2	g=3	g=4	g=5	g=6	g=7	g=8
k=0	f(0,g) =	0	0	0	0	0	0	0	0	0
k=1	f(1,g) =	0	**15**	15	15	15	15	15	15	15
k=2	f(2,g) =	0	**15**	15	15	15	15	25	25	25
k=3	f(3,g) =	0	15	15	15	**24**	24	25	25	25
k=4	f(4, g) =	0	15	15	15	24	24	25	25	**29**

Last value: k=n, g=M

$f_{max} = f(n,M) = f(4,8) = 29$

Backtracking the solution:

Repeat for k =n ,n-1 ,..., 1
If $f(k,g) \neq f(k-1,g)$, item k is in the selection, $x_k := 1$. Otherwise, $x_k := 0$.
Capacity for previous items: $g := g - w_k x_k$

g=8
k=4: $f(4,8) \neq f(3,8) \Rightarrow x_4 = 1$
$g = g - w_4 = 8-4 = 4$

k=3 $f(3,4) \neq f(2,4) \Rightarrow x_3 = 1$
$g = g - w_3 = 4-3 = 1$

k=2, : $f(2,1) = f(1,1) \Rightarrow x_2 = 0$
$g = g - 0 = 1$

k=1 $f(1,1) \neq f(0,1) \Rightarrow x_1 = 1$
$g = g - w_1 = 1-1 = 0$

The solution is x = (1,0,1,1) i.e. items 1,3, and 4 are selected. value of the knapsack is 29.