Define how Knapsack Problem is Solved by dynamic programming.

Consider n=3(w, w, w_1)=(2, 3, 3), (P,, P_2, P_2)=(1,2,4) and 6. Find optimal solution.

n = 3 (number of items)

w_1 = 2, w_2 = 3, w_3 = 3 (weights of the items)

P_1 = 1, P_2 = 2, P_3 = 4 (values of the items)

W = 6 (maximum weight capacity)

Step 01:

Create table.

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0						
2	0						
3	0						

Step 02:

Fill in the table:

For item 1 (w_1 = 2, P_1 = 1):

	0	1	2	3	4	5	6

0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1
2	0						
3	0						

Step 03:

For item 2

($w_2 = 3$, $P_2 = 2$):

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1
2	0	0	1	2	2	2	2
3	0						

Step 04:

For item 3

($w_3 = 3$, $P_3 = 4$):

	0	1	2	3	4	5	6

0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1
2	0	0	1	2	2	2	2
3	0	0	1	4	4	4	6

The optimal solution is the value in the bottom-right corner of the table, which is dp[3][6] = 6. Therefore, the maximum value that can be achieved without exceeding the weight capacity of 6 is 6.