Lab 10 Dynamic Programming

Q1.
Consider the Knapsack problem algorithm we covered in class i.e.

$$V[i,j] = \begin{cases} \max\{V[i-1,j],\ v_i + V[i-1,j-w_i]\} & \text{if } j-w_i \geq 0 \\ V[i-1,j] & \text{if } j-w_i < 0 \end{cases}$$

Initial conditions: $V[0,j] = 0$ and $V[i,0] = 0$.

This does not list the items selected when a final optimized solution is obtained.

Use this algorithm and modify it by adding a new array Keep[i,w] which will keep track of the items selected. Write a pseudo code for this updated algorithm. Also, make sure you print the results showing items selected [Hint: use similar approach as in Print-Cut_Rod_Soln].

Q2.
Determine the running time complexity of your pseudo code. Has Dynamic Programming approach of solving Knapsack problem changed the exponential time complexity of the original brute force solution?

Q3.
Use the answer of Q2 to explain how Knapsack problem is exponential (Hint - consider both inputs and see how they [or at least one] can cause Knapsack problem to be exponential).