1. Consider a modification of the rod-cutting problem in which, in addition to a price p_i for each rod, each cut incurs a fixed cost of c. The revenue associated with a solution is now the sum of prices of the peices minus the cost of making the cut.

   (a) Give a dynamic-programming algorithm to solve this modified problem, including the mathematical expression for the maximum revenue and the pseudocode.

   (b) Show the maximum revenue r_j and the optimal size s_j of the first piece to cut off, when $c = 1$ and

length i	1	2	3	4	5	6	7	8	9	10
price p_i	1	4	5	9	10	12	15	18	19	20

2. Find an optimal parenthesization of a matrix-chain product $A_1A_2A_3A_4A_5$, whose sequence of dimensions is $p = (5, 6, 3, 7, 5, 3, 4)$.

   (a) Compute the dynamic programming m-table and s-table.

   (b) Show the optimal parenthesization.

3. For the sequences $X = \langle B, C, A, A, B, A \rangle$ and $Y = \langle A, B, A, C, B \rangle$,

   (a) Follow the pseudocode LCS-LENGTH to fill in the dynamic programming c- and b-tables for finding the longest common subsequence (LCS) of X and Y.

   (b) Follow the pseudocodes PRINT-LCS, list the LCS.

4. Two character strings may have many common substrings. Substrings are required to be contiguous in the original string. For example, *photograph* and *tomography* have several common substrings of length one (i.e., single letters), and common substrings *ph*, *to*, and *ograph* (as well as all the substrings of *ograph*). The maximum common substring (MCS) length is 6.

   Let $X = x_1x_2 \cdots x_m$ and $Y = y_1y_2 \cdots y_n$ be two character strings.

   (a) Give a dynamic programming algorithm to find the MCS length for X and Y.

   (b) Analyze the worst-case running time and space requirements of your algorithm as functions of n and m.

   (c) Demonstrate your dynamic programming algorithm for finding the MCS length of character strings *cabccb* and *babcba* by constructing the dynamic programming tables.

5. This problem continues Problem 6 of Homework #4 to solve the following 0-1 knapsack problem by using dynamic programming:

   Given six items $\{(v_i, w_i)\}$ for $i = 1, 2, \ldots, 6$ as follows:

i	v_i	w_i
1	40	100
2	35	50
3	18	45
4	4	20
5	10	10
6	2	5

   and the total weight $W = 100$, where v_i and w_i are the value and weight of item i, respectively. Compute the solution by dynamic programming and comment your findings.

   You need to outline your DP algorithm, not just given the answer. Since W is pretty big, you need to write a short program to do the calculation and submit a hardcopy of your program.