Max Rectangular Sum Problem

Max rectangular problem is to find the “rectangle” having the max sum of elements in it on a given table. For example, in the following example the rectangle defined by the shaded area has the sum value of -45.

12	5	-2	7	-80	0	-5	7
10	20	15	5	20	-5	8	-20
5	-90	10	17	8	3	15	1
8	14	2	0	1	11	10	0

The solution of the problem for this table is given below.

12	5	-2	7	-80	0	-5	7
10	20	15	5	20	-5	8	-20
5	-90	10	17	8	3	15	1
8	14	2	0	1	11	10	0

Solve this problem following the steps below:.

a) Write a function to read all doubles from a text file into a 2D array having 8 columns. Your function should return number of lines it read as an output parameter.

b) Consider a simplification of this problem, in which the left upper coordinate of the rectangular is constant. The problem is to find max sum using any rectangular, whose upper corner is given. For example if (0, 0) point is given for the upper corner, the result for the table below is the sum of the elements in the shaded area.

12	5	-2	7	-80	0	-5	7
10	20	15	5	20	-5	8	-20
5	-90	10	17	8	3	15	1
8	14	2	0	1	11	10	0

Write a function solving this problem. The function should take the table, number of its rows and upper left coordinates of the rectangular, return the max sum as the return value and right below coordinates of the rectangular of the solution as the output arguments with the following prototype:

double maxRecSumConstPoint(double table[][8], int nRows, int leftUpXCoor, int leftUpYCoor, int* rightDownXCoor, int* rightDownYCoor)

Write a function to calculate the sum inside any rectangular as a helper function. The function should take the left above and right below coordinates of the rectangular and return the sum.

c) Write a function to solve the problem without any simplifications. This function should call “maxRecSumConstPoint” for each point in the table as the left upper corner and find max possible rectangular sum.