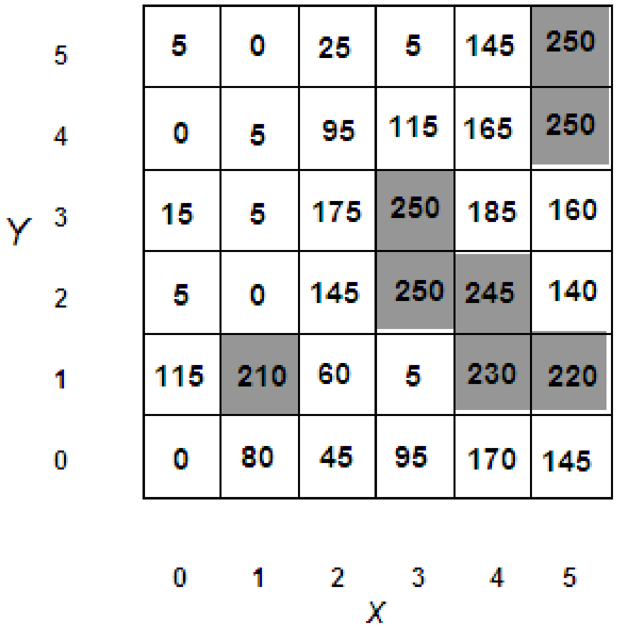
Problem Background: Given as input a two-dimensional array of values (that represent, say, measurements of some signal strength over each cell in a region), we wish to automatically generate a description of the ***sub-regions of interest*,** defined as all contiguous groups of adjacent cells for which the signal is greater than a threshold value ***T***. Cells that are touching at corners are considered to be (diagonally) adjacent cells.

For example, consider the 6 x 6 array shown in the following diagram. If we have threshold value *T* = 200, then the three gray-shaded sub-regions identified are the sub-regions of interest. The sub-regions of interest can be identified by listing the X,Y coordinates of the cells they contain. In the example, these would be:

- Lower-left sub-region: { (1,1) }
- Middle "W-shaped" sub-region: { (3,3), (3,2), (4,2), (4,1), (5,1) }
- Upper-right sub-region: { (5,4), (5,5) }



	5	5	0	25	5	145	250
	4	0	5	95	115	165	250
Y	3	15	5	175	250	185	160
	2	5	0	145	250	245	140
	1	115	210	60	5	230	220
	0	0	80	45	95	170	145
		0	1	2	3	4	5
					X		

Problems

If any necessary details are omitted from the problem descriptions below, you should define whatever additional information is needed. Please document these additions, along with any significant assumptions you make.

1. Write down a function, expressed either in C# or C++, that takes two inputs:
   - a two-dimensional array of values
   - a threshold T

   and determines this output:
   - A list of (X,Y) value pairs that identify the *center of mass* of each region of interest. (There will be one X,Y coordinate pair for each sub-region.) Define "center of mass" for a given sub-region to be the average position, expressed in X,Y coordinates, of the cells in that sub-region, each cell's location being weighted by that cell's signal value.

Feel free to express your response as compilable code. (If you do, please include any make or project files.)

2. Discuss a preferred strategy for testing the function you defined in part 1, including how you might automate the testing, how you would select test input cases (feel free to enumerate cases), and how you would determine whether each test case results in a pass or fail.