COMP-10205 – Data Structure and Algorithms

Assignment#1
Mountain Peaks and Valleys

Arrays and Multi-dimensioned Arrays
5% of course grade

Submission Requirements

Complete the following exercise and submit electronically in the assignments folder on eLearn as an IntelliJ Project – Zip the entire folder not just the source files in MyCanvas. Please refer the course Calendar for the exact date and time of the submission. This assignment is to be completed individually.

Background

You are to develop a program that will analyze a set of elevation data collected in a text file. The text file is formatted as a series of rows and columns. All the values in the data file are integer values in the range of 20000 to 112000 inclusive. However, the range of data could be smaller than the absolute range. See the Assignment#1 folder for a copy of the data file called ELEVATIONS.TXT.

The first row in the data file contains four numbers:
1) Number of Rows, 2) Number of Columns, 3) Minimum Peak Height, 4) Exclusion Radius (used for peak detection).
The data starts at line 2 of the data file and each value is separated with a space.

Requirements

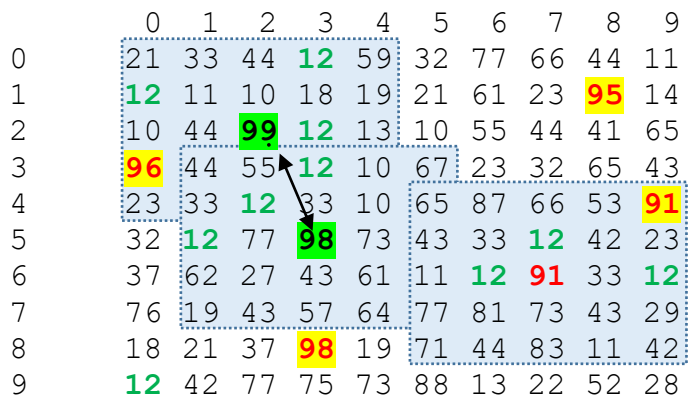
You must determine the following about the data set:

1. Print the lowest elevation value and the number of times it is found in the complete data set.

	0	1	2	3	4	5
0	21	33	44	12	59	32
1	12	11	**10**	18	19	21
2	**10**	44	99	12	13	**10**
3	96	44	55	12	**10**	67
4	23	33	12	33	**10**	65
5	32	12	77	99	73	43

   You will notice that 10 is the lowest value in the data set and it occurs 5 times.

2. Print all the local peaks where the peak elevation is greater or equal to 111300 using an exclusion radius(n) of 13. A local peak occurs when the value it is greater or equal to 111300 and is higher than all the values within n rows and columns inclusive in all directions of the data set. Local peaks in the first n rows, first n columns, last n rows and last n columns can be excluded from the analysis. The example below is a 10 x 10 data set where the exclusion radius (n) is set to 2 and where we are interested in all local peaks greater or equal to 90



	0	1	2	3	4	5	6	7	8	9
0	21	33	44	12	59	32	77	66	44	11
1	12	11	10	18	19	21	61	23	95	14
2	10	44	99	12	13	10	55	44	41	65
3	96	44	55	12	10	67	23	32	65	43
4	23	33	12	33	10	65	87	66	53	91
5	32	12	77	98	73	43	33	12	42	23
6	37	62	27	43	61	11	12	91	33	12
7	76	19	43	57	64	77	81	73	43	29
8	18	21	37	98	19	71	44	83	11	42
9	12	42	77	75	73	88	13	22	52	28

There are 2 local peaks which occur are 99(2,2) and 98(5,3). Note that although 91, 95, 96 and 98 are examples of local peaks they occur in the excluded area (first and last 2 rows, first and last 2 columns). Also note that the value of 91(6,7) is not considered a local peak as it is the same elevation as another value within the exclusion radius - they cancel each other out (must be greater not equal.

3. Print the **row and column and elevation** of the two closest local peaks using the formula for distance presented below:

$d^2 = (r1 - r2)^2 + (c1 - c2)^2$
where d is the distance
r1,r2 are the row numbers of the two peaks
c1,c2 are the column numbers of the two peaks

In the data presented above the min distance would be **3.16** (the two peaks located at (2,2 with an elevation of 99) and (5,3 with an elevation of 98). Print the distance between the two peaks to 2-decimal places.
Challenge: There may be more than one set of closest peaks – can you find out how many and print all of them.

4. Print the most common elevation in the data set. In the data set presented above the most common value is 12 and it occurs 10 times.

Your solution must read all the data into a two-dimensional array and then close the file. Each of the 4 parts above **MUST be solved by writing a single method (1 for each part) that is passed the two-dimensional array and any other input you desire**. The method may return data of any type (returning arrays can be quite useful here). Also note that if you save all the Peaks in Part 2, you can use this data for Part 3. You may wish to create a class to store the Peaks.

Your solution must use only arrays and multi-dimensioned arrays. No array lists or other data structures are allowed for this problem. You may not sort the data

Marking Scheme

Program Structure – Implemented using best practices and arrays only, no sorting / commented - 35%
All options implemented, complete and correct – 50%
Efficiency (completes in less than 1 second – start timing after data has been read from file) – 15%