**WRAV202: Practical 6**

**Note from your lecturer:** *I have noticed that on my office PC the line breaks do not show up in the text files provided with this practical when I use* Notepad *(all the text just appears on a single line). The line breaks do, however, appear when I open the files in* WordPad*. This is weird and I hope that it does not cause you too much confusion.*

# Question 1

For this question, you must write a program that can create binary trees. Your program will build binary trees from the bottom up. The input file shows the contents of the tree in layers from left to right, where each line is a layer. Each node on each line is separated by a comma. The trees are all perfectly balanced, so all the nodes per level will be present in the file. Below is an example of how the input file is converted into a binary tree:

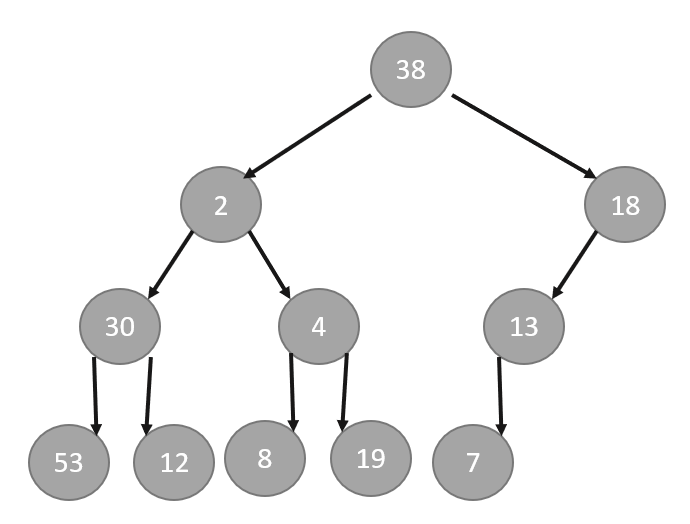
Input file:

53,12,8,19

30,4

2

Resulting tree:



There are multiple trees in the input file, and each time a line has the text **<new tree>**, then it is a new tree that needs to be processed.

After each tree is created, your program must display the trees in pre-order, post-order and level-order and write it to an Outputs.txt file. Each tree will produce three lines of output in the Outputs.txt file. The first line is the pre-order, the second is the post-order, and the third is level-order. Use commas as separators between nodes. In the case above, the output is:

2,30,53,12,4,8,19

53,12,30,8,19,4,2

2,30,4,53,12,8,19

# Question 2

Load the trees from the input file as done previously.

Your program must find the minimum value, the maximum value in each tree and the sum of all of the values in each tree. Write them to the output file in this order: min,max,sum

Each line in the output file represents a new tree.

# Question 3

Load the trees from the input file as done previously.

Your program must find the average of the values in each tree and write each average to a new line in the output file. Round the average to two decimal places.

# Question 4

Load the trees from the input file as done previously.

Your program must find the sum of the values in all **internal** nodes of each tree and write it to a new line in the output file.

# SUBMISSION

You can complete each task as a separate method in a one program. Submit this program as a single file with a *.cs* extension.