Modeling Challenge III: Calibron 12

Eckel, TJHSST AI1, Fall 2019

# Background & Explanation

To finish off our unit on constraint satisfaction, you will be solving the Calibron 12. More specifically, you’ll be solving a variety of Calibron-12-like puzzles; the Calibron 12 itself is *quite* challenging!

# Required Task

You will be given the size of the containing rectangle and then the dimensions of a number of smaller rectangles. The following input string specifies the Calibron 12 puzzle itself:

"56 56 28x14 32x11 32x10 21x18 21x18 21x14 21x14 17x14 28x7 28x6 10x7 14x4"

The first number is the height of the puzzle, the second the width. After that are the dimensions of each rectangle. Note rectangles can be repeated, and note also rectangles can be rotated (so 21x18 might be 18x21 in the solution.)

There is a tiny amount of shell code (really just code to read in and unpack the puzzle statement) that also contains specific instructions about input and output formatting. Be sure you read them carefully. ***Note, please, that your code will receive one puzzle (a single long argument string in quotes) directly on the command line, not a filename as usual.***

You can solve this puzzle any way you want. There are no rules about data structure or method (except for the usual stipulations that you can’t google a solution or import any non-standard libraries). It just has to work.

The output format is specified in the shell code.

# Test Cases

This test case should output “Containing rectangle incorrectly sized.”

"10 13 3x6 2x5 4x10 7x9 1x1"

This test case should output “No solution.”

"8 5 4x2 4x2 4x2 4x2 2x4"

Here are some standard test cases. To get full credit on this assignment, you should be able to solve each of the following test cases in no more than 15 seconds per test case:

"4 7 7x4"

"2 3 1x2 2x2"

"18 9 3x11 5x7 4x8 6x10 1x2"

"4 8 4x1 1x6 1x3 3x1 1x3 1x3 6x1 1x4"

"11 12 3x6 2x5 4x10 7x9 1x1"

"9 18 3x8 5x10 4x11 6x7 1x2"

"13 14 4x5 3x8 6x11 7x10 2x1"

"19 19 1x19 1x12 6x9 9x15 15x3 10x6 3x12"

Here are a couple hard cases. To get outstanding work credit on this assignment, you should be able to solve each of the following test cases in no more than 1 minute per test case (in particular, the final test case is the actual Calibron 12):

"24 24 4x7 7x4 7x5 11x4 9x3 2x8 22x3 8x4 14x11 8x10 2x12 5x3 9x3"

"56 56 28x14 32x11 32x10 21x18 21x18 21x14 21x14 17x14 28x7 28x6 10x7 14x4"

As with the Rush Hour assignment, you can also declare failure. You must state your intentions to declare success or failure at the end of the class period prior to the due date.

# Specification if you Declare Success

If you believe you have succeeded in the task, submit a single Python script to <https://tinyurl.com/F19EckelCalibron12>.

This assignment is **complete** if:

* The “First Name” field on the Dropbox submission form contains your **class period**, not your name.
* The “Last Name” field on the Dropbox submission form contains your **last name then a comma then your first name** (like, for example, “Eckel, Malcolm”).
* Your code does all of the following:
  + Accept a set of command line arguments specifying one puzzle.
  + Output a solution to the puzzle according to the specification in the shell code. (Please follow it **exactly**. Any extraneous print statements, including any empty lines, will cause my grader to fail.)
* Total runtime is less than 2 minutes. See timing hints on the front to check yourself.

# Specification if you Declare Failure

If you decide that your code can’t be completed, I’ll need a write-up turned in to me (printed out; turned in by hand.)

The assignment is **complete** if:

* You write a 500 – 1000 word analysis of your code including what you tried to do, what the code actually does, and why it isn’t working. It should be printed out and ready to turn in when you walk in the room.

# Specification for Outstanding Work

Submit a single Python script to <https://tinyurl.com/F19EckelCalibron12Fast>.

The specification is identical to the “Declare Success” specification above except for the different submission link and the difficulty of puzzle your code will need to solve. You will need to solve the actual Calibron 12 (no pre-stored solutions, of course – I’ll be checking) in addition to other test cases of similar difficulty. See timing hints on the front to check yourself.

Due to the time-limited nature of this assignment (and the fact that I will be handing out solution code), **this Outstanding Work specification must be turned in by the same deadline as the normal assignment and cannot be completed late.**