COS 301 Spring 2024 HW03 [1] *Points: 100. Pages: 1.* *Due 2024-03-06 09:05 a.m.*

Please follow the submission procedure from earlier homeworks, replacing the appropriate tags with `hw03`. The rules for using outside resources are also similar to those used earlier, as are other guidelines on code and input-output. As before, use of the class discussion forum for elucidating the details is expected and strongly encouraged.

This homework centers on **further extending the `calc.py`** example (beyond what was done in the previous homework) as follows.

1. In addition to scalar variables and literals, it should support lists of scalars. (Lists of lists of scalars and so on do not need to be supported.) The list constructor uses Python-*like* syntax. For example, if `x` has value 1, then `(3, 1, 4)` and `(3, x, 4,)` (note also the trailing comma) construct the identical lists, composed of the three integers 3, 1, 4 in that order. The motivation for the optional terminal comma is similar to that in Python. In the output, a list should be represented in the second form (including trailing comma) with exactly one space after each internal comma. The calculator uses *duck typing* like Python, so the same variable may store a scalar and a list at different times.
2. All operators (including those in the earlier extension) and the assignment statement should be extended to lists. A unary scalar operator is extended to a list by applying it element-wise to the list's elements. A binary operator is similarly extended to a pair of list operands by applying it to the corresponding (in order) pairs of the lists' elements.
3. If a binary operator is applied to lists of unequal lengths then the last element of the shorter list is conceptually repeated to make the lengths equal. If the shorter list is empty then 0 is used if the operator is addition or subtraction and 1 if it is multiplication or division.
4. If one of the operands of a binary operator is a list and the other a scalar, then the scalar is promoted to a singleton list (after which the previous rule may be applied if necessary).

Sample inputs and outputs should be discussed on the class discussion forum.

Additional submission requirements (beyond those in earlier homeworks):

- Your packaged submission must include 10–20 sample inputs and corresponding outputs. You should select these inputs carefully with the goal of showcasing your program's capabilities. Descriptions of the inputs and outputs that highlight why they are interesting should be included in the README file.

[1] This version does not differ from the early-acess version posted earlier except for this footnote.