**Elyah Coding Challenge Design Document**

Language: Python 3

Data Structures: A LIFO data structure (stack) was chosen for this problem since the “backtracking” functionality is particularly useful for computing the order of mathematical operations. While an array or list would have also sufficed for this problem, their flexibility is not needed and they are also easily corruptible.

* Stack for operands: keeps track of operands of the expression as they are traversed
* Stack for operators: a separate stack for operators, keeping track of “+-\*/()” as they are traversed

Algorithm Approach

* Helper Functions
  + Compute a value given two operands and an operator in the form of characters
  + Assign a priority to an operator based on the order of operations
* Primary Function
  + Initialize the stacks for the operands and operators
  + Remove all whitespace from the input string
  + Traverse each character of string:
    - If character is an open parenthesis, push it to the operator stack
    - If character is a digit, decipher the entire operand value by traversing until non-digit is found
      * Account for negative (-) character and/or decimal (.) character in the deciphering of the operand value
    - If character is a close parenthesis, evaluate all the operations within the parentheses in the order of operations, while updating the operand stack accordingly
    - If character is an operator, compute all operations before current operator of higher or equal priority
  + Perform all remaining computations after the string has been traversed and return the result from the top of the operand stack