Liam Chapman – COSC326 Etude 10 – Arithmetic

Compiles with: javac Arithmetic.java

Runs with: java Arithmetic < tests.txt

After parse checking to make sure the input is correct, I do a loop to check the possible max and min values with the given numbers, if the target is lower than the min or higher than the max, fail.

My left to right is a simple recursive method which takes in the sum, a string which constructs the sequence as it goes and the index(for the array of numbers). Firstly it checks if the target == sum and index == the amount of numbers, if so correct! Then it to make sure the sum is less than or equal to (incase the remaining numbers are 1) the target and that the index is less than the number of numbers. In this check is just has 2 calls to itself, one being passing through sum + number[index] and the other the same with times, changing the string sequence and increasing index.

My normal Bedmas method is the exact same as the left to right apart from the recursive multiplying calls, the String sequence and a new parameter prevTimes. First the string sequence has been changed to a string or +'s or *'s to help as a good reference to which operand was used at any given point (not needed for left to right). With the multiply recursive calls, I first check is the length of the sequence is 0 (a.k.a. nothing has been done yet), if so then prevTimes = first value * 2nd value, and recurse with prevTimes as the sum and itself (prevTime will always recurse as itself in the multiply calls). Secondly if the previous operand was a +, if so prevTimes = index-1 number * current number, and the sum calls through as itself minus the index-1 number. Lastly if the previous operand was a * sum will equal itself minus prevTimes, prevTimes will have itself timesed by the current number(or if it was 0 by 1) and then called with sum + prevTimes.