Methodology

Section 1: getInput()

This section handles retrieving and parsing data from the user. Input is taken using a Scanner object and stored into 2 strings.

Section 2: getProduct()

This sections is broken down into two subsections, discussed later, with the overarching goal of finding the product of the numbers by finding and summing intermediates.

Section 3: printResults()

This section simply prints out the results, trivial considering that the results are calculated and have been stored prior to using this method.

Section 4: printIntermediates()

This section prints out the intermediates by popping them from the stack and printing until the stack is empty.

Helper Section 1: getIntermediates()

This section finds all intermediate products of the multiplication with nested loops that traverse the number strings, starting at the back, and multiplying each character and adding a carry. The carry is adjusted when the result is greater than 10 and the result is also adjusted. After this the result is added to the front of the current intermediate string. At the end of the inner loop any remaining carry is added to the front of the intermediate, then the carry is reset and the intermediate is pushed to the top of the intermediateStack. Then the current intermediate is reset and filled with zeroes depending on the current iteration. Finally, the loop counter is decremented and the loop continues.

Helper Section 2: sumIntermediates()

This section sums the values of the intermediate stack, it does this by taking the first value from the intermediate stack and storing it as the product, then while the intermediate stack has values, elements are popped and added to product, padding is added when needed. Every time an element is popped from the intermediate stack it is added to the final stack. After the loop is finished every element from intermediate stack is now in final stack and product is the sum of all intermediates.

Helper Section 3: padString(s, length)

This section will loop until the length of the string, s, is equal to the target length, length. This is done with a simple loop, in the loop a ‘0’ is added to the front of the string. Finally, the padded string is returned.

Helper Section 4: sumNumbers(s1, s2)

This section will loop through both string, starting at the end, and at each character and carry together to find the result. The result and carry are adjusted and the result is joined to the front of the sum string. When the loop has ended, any remaining carry is added to the front of sum and sum is returned.

Proofs