Next we add the three decimal digits in the tens place, 1 (the carry into the tens place from the ones place) + 6 + 7. The sum of these three digits exceeds 10 by 4, which we show by placing a 4 in the tens place in the sum and recording the fact that there is an ultimate carry of one. Recall that we had decided to use only two digits, so there is no hundreds place. Using the notation of Equation (2.3.3), we describe addition of two decimal integers in Algorithm 3.1.1.

Algorithm 3.1.1 Add Fixed-Width (N-Digit) Integers in the Decimal Number System.

$carry_0 = 0$

For $i = 0, \cdots, (N-1)$

$sum_i = (x_i + y_i + carry_i) \ \% \ 10$

$carry_{i+1} = (x_i + y_i + carry_i)/10$

Note that:

- Algorithm 3.1.1 works because we use a positional notation when writing numbers—a digit one place to the left counts ten times more.
- Carry from the current position one place to the left is always 0 or 1.
- The reason we use 10 in the `/` and `%` operations is that there are exactly ten digits in the decimal number system: 0, 1, 2,..., 9.
- Since we are working in an N-digit system, we must restrict our result to N digits. The final carry, $carry_N$, is either 0 or 1 and is part of the result, along with the N-digit sum.

Let us now turn to the subtraction operation. As you recall from subtraction in the decimal number system, you must sometimes *borrow* from the next higher-order digit in the minuend. This is shown in Algorithm 3.1.2.

Algorithm 3.1.2 Subtract Fixed-Width (N-Digit) Integers in the Decimal Number System. *Subtracting y from x.*

$borrow = 0$

For $i = 0, \cdots, (N-1)$

If $y_i \leq x_i$

$difference_i = x_i - y_i$

Else

$j = i + 1$

While $(x_j = 0)$ *and* $(j < N)$

$j = j + 1$

If $j = N$

$borrow = 1$

$j = j - 1$

$x_j = x_j + 10$

While $j > i$

$x_j = x_j - 1$

$j = j - 1$

$x_j = x_j + 10$

$difference_i = x_i - y_i$