### Counting and Summing the Digits of an Integer

The problem of counting the digits in a positive integer or summing those digits can be solved recursively as follows:

* If the integer is less than 10 there is only one digit (the base case).
* Otherwise, the number of digits is 1 (for the unit’s digit) plus the number of digits in the rest of the integer. For example, the number of digits in 3278 is 1 + the number of digits in 327.

Here is the recursive algorithm implemented in Java.

public int numDigits(int num) {

if (num < 10)

return 1; // a number < 10 has only one digit

else

return 1 + numDigits (num / 10);

}

Note that in the recursive step, the value returned is 1 + the result of the call to determine the number of digits in *num/10*. Recall that *num/10* is the quotient when *num* is divided by 10 so it would be all the digits except the units digit.

Write a program called DigitPlay as follows:

* Copy the numDigits method as per above. Make sure it is a static method because you will be calling it from a static context.
* In the main method, ask for a number from the keyboard, send it to you numDigits method, and when it returns, print the result.
* Now write another static method, called sumDigits, that sums the digits of a number recursively. The algorithm should be similar to counting the number of digits, except it should add the actual digits.
* To test it, pass the same number from the keyboard in the main method, and print the result when it returns.