

Counting and Summing Digits in an Integer

The problem of counting the digits in a positive integer or summing those digits can be solved **recursively**. For example, to count the number of digits think 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 units digit) plus the number of digits in the rest of the integer (what's left after the units digit is taken off). For example, the number of digits in 3278 is 1 + the number of digits in 327.

The following 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 (counts the units digit) + the result of the call to determine the number of digits in $\text{num} / 10$. Recall that $\text{num} / 10$ is the quotient when num is divided by 10 so it would be all the digits except the units digit.

The file **DigitPlay.java** contains the recursive method **numDigits** (note that the method is static -- it must be since it is called by the static method **main**). Copy this file to your directory, compile it, and run it several times to see how it works. Then modify the program as follows:

Add a static method named **sumDigits** that finds the sum of the digits in a positive integer. Also add code to the main method to test your new method. The algorithm for **sumDigits** is very similar to **numDigits**; you only have to change two lines.