Horner's Algorithm

Our first pass at describing Horner's algorithm will use English as much as possible:

Input: A integer R which is greater than one.
A sequence of digits in a base specified by the input R.
Output: The base 10 equivalent of the input sequence.

Method:

```
Start with the first digit as the working number
repeat
    multiply the working number by the radix, R
    add the next digit
until there are no more digits
ouput the working number
```

In a high-level computer language, we must specify our algorithms in terms of operations that are supported by that language. In turn, the language features are chosen to be those which can be supported by the underlying hardware. Here is another description of Horner's algorithm in our pseudocode notation.

Input: A integer R which is greater than one.
A sequence of digits in a base specified by the input R.
Output: The base 10 equivalent of the input sequence.

Method:

```
t = 0                    // Start with nothing.
loop = true              // Use a loop control variable
while (loop) {
    read d               // Read the next digit
    if ( the read was successful ) {
        t = t * R + d
    }                        // end if ...
    else {
        loop = false
    }                        // end else ...
}                        end while ...
output t
```