The expectation that the modification would run twice as fast is not entirely confirmed, with the runtime for the primes being a bit over half of what it was before. Were it truly twice as fast, we would expect an improvement of around 2.00 below, where 3 runs of testing the 3 lowest primes after the number listed in the **Prime** column were averaged below for both the old and new version of `smallest divisor`.

Prime	Old	New	Improvement
10^{10}	71.556	44.556	1.606
10^{11}	236.556	142.667	1.658
10^{12}	728.778	415.556	1.754

The reason that the observed ratio is different from 2 is because the number of steps is not exactly half of what it was before. In fact, is was greater than half, which is why the runs were slower than expected. This is because we added an extra procedure call (`next`), which itself included an `if` procedure and resulting predicate (`= n 2`), all of which are repeated for each `test-divisor`.

Stripping the procedure call and placing the if statement directly in the `else` clause improves marginally for all but 10^{12}, where it is about the same.

Prime	Old	Remove `else` Call	Improvement
10^{10}	71.556	41.333	1.731
10^{11}	236.556	133.889	1.767
10^{12}	728.778	416.333	1.75

What if we only test whether the `test-divisor` is equal to 2 once? This would greatly reduce the number of unnecessary predicate evaluations, because once we have a `test-divisor` greater than 2, it will never be equal to 2 for the remainder of that `n`, since `test-divisor` can only be increased by the `smallest-divisor` procedure. To implement this, the first time we will call `find-divisor` which will check if the initial test-divisor, 2, is greater than `n` when it is squared (4), and whether or not 2 (and thus any even number) divides `n`. If neither of these are the case, `find-divisor-odds` will be called which no longer has to check if `test-divisor` equals 2.

Prime	Old	Check Once	Improvement
10^{10}	71.556	39.667	1.804
10^{11}	236.556	125.111	1.891
10^{12}	728.778	388.333	1.877

This finally yields something closer to the 2.000 improvement that we would expect.