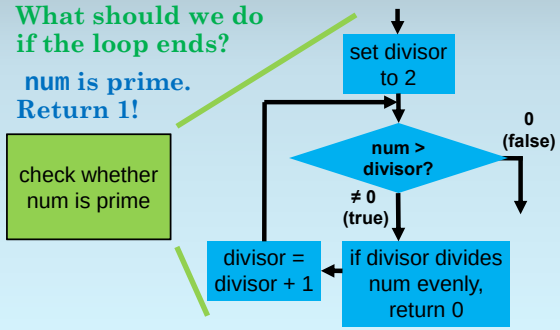
What's True if the Loop Ends?

What should we do if the loop ends?

`num` is prime. Return 1!



Stop When We Can Write as C Functions

We can check whether numbers divide evenly with a C function!

But we need a function signature…

```
int32_t divides_evenly
   (int32_t divisor, int32_t value);
// Returns 1 if divisor divides
// value evenly, or 0 otherwise.
```

Now we're ready to write `is_prime`.

Our `is_prime` Function for Checking Primality

```
int32_t is_prime (int32_t num)
{
    int32_t divisor;
    for (divisor = 2; num > divisor;
         divisor++) {
        if (divides_evenly
            (divisor, num)) {
            return 0;
        }
    }
    return 1;
}
```


Use Integer Arithmetic to Test for Multiples

For integers **A** and **B**, what does the expression **(A / B) * B** produce?

The largest multiple of B that is not more than A.

Let's use this expression to write **`divides_evenly`**.*

*Equivalently, one can just use modulus.

