Still Need to Write the Helper Function

That's it for allocation.

But what did this do?

```
bin = log2_ceil (block_size);
```

Calculate k such that $2^k \geq$ `block_size`.

In other words, **return $\lceil \log_2(\text{block\_size}) \rceil$** (the ceiling of the base 2 logarithm).

How can we calculate that value?

How Can We Calculate Ceiling of Log_2?

```
// Returns ceiling of
// log_2 of its argument.
static int32_t log2_ceil
    (size_t value);
```

One option: library calls (with floating-point).

Instead, **let's use...**

bits!

Find the First 1 Bit and Check for a Power of Two

Let's look at a number as bits:

```
value = 000…000 1 ??????
```

To **calculate ceil ($\log_2$ (value))**, we
- **find** the location of **the first 1 bit**, and
- **round up** unless all of the lower bits are 0.

Let's start with the second part.

How can we check: is `value` a power of 2?

Initialize Count to Reflect Whether `value` is a Power of 2

```
static int32_t log2_ceil
    (size_t value)
{
    int32_t ret_val;
    if ((value & (value - 1)) == 0){
        ret_val = -1;
    } else {
        ret_val = 0;
    }
```

Is **value** a power of 2?

If so, start counting at -1.

If not, start counting at 0.

