Calculate Necessary Values and Check Arguments

```
block_size = n_bytes +
             sizeof (*new_block);
if (n_bytes == 0 ||
    block_size > MEM220_MAX_ALLOC) {
    return NULL;
}
bin = log2_ceil (block_size);
```

Size too small or too large? Give up.

Calculate Necessary Values and Check Arguments

```
block_size = n_bytes +
             sizeof (*new_block);
if (n_bytes == 0 ||
    block_size > MEM220_MAX_ALLOC) {
    return NULL;
}
bin = log2_ceil (block_size);
```

Find the right bin (function discussed later).

Two Places to Obtain a Block

Does the right list have a free block in it?

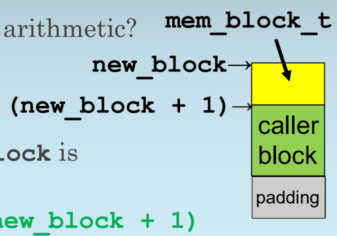
```
if (mem_bin[bin] != NULL) {
    // get block from free list
} else {
    // allocate a new block
}
return (new_block + 1);
```

Both cases set **new_block**.

What's this?

Pointer Arithmetic Gives the Right Answer

Remember pointer arithmetic?



The type of **new_block** is **mem_block_t***.

So where does (new_block + 1) point?

To the block to be returned!