5) Explain what the following code does: `((n & (n - 1)) == 0)`.

If `(n & (n - 1)) == 0`, it means that there is no bit position in n and n – 1 which shares a 1 in both numbers.

When we subtract 1 from a binary number, if there is a 1 in the units place, it becomes 0. If not, we borrow a 1 from the least significant 1 and all the 0's before that 1 now become 1. The digits before the least significant 1 remain unchanged.

So, for `(n & (n - 1))` to be 0, in n, there should be no 1 to the right of the least significant 1. In other words, it must be a number with a 1 followed by any number of 0s. Such a number is a power of 2.

Thus, `(n & (n - 1)) == 0` checks if a number is a power of 2 (or is 0).