The formula:

```cpp
int digits = log10(n) + 1;
```

is used to count the number of digits in a positive integer `n`. Here's why it works:

Understanding Logarithms

The **base-10 logarithm** (`log10`) of a number `n` gives the power to which `10` must be raised to obtain `n`.

$$\log_{10}(n) = x \quad \text{such that} \quad 10^x = n$$

For example:

- $\log_{10}(1000) = 3$ because $10^3 = 1000$
- $\log_{10}(543) \approx 2.73$ because $10^{2.73} \approx 543$

Derivation of the Formula

If `n` has `d` digits, it falls within this range:

$$10^{(d-1)} \leq n < 10^d$$

Taking the logarithm on both sides:

$$\log_{10}(10^{(d-1)}) \leq \log_{10}(n) < \log_{10}(10^d)$$

Simplifying:

$$d - 1 \leq \log_{10}(n) < d$$

Now, adding `1` to both sides:

$$d \leq \log_{10}(n) + 1 < d + 1$$

Since `d` is an integer, taking the **floor value** of `log10(n) + 1` gives `d`.

Example Calculations

1. **For** `n = 1000`
   - $\log_{10}(1000) = 3$
   - $3 + 1 = 4$ → Correct, `1000` has **4 digits**.
2. **For** `n = 543`
   - $\log_{10}(543) \approx 2.73$
   - $2.73 + 1 = 3.73$
   - Floor value → `3` digits.

Edge Cases

- `n = 1` → $\log_{10}(1) = 0$, so `0 + 1 = 1` (correct).
- `n = 999` → $\log_{10}(999) \approx 2.99$, so `2.99 + 1 = 3.99` (floor is `3` → correct).
- **If** `n = 0`, `log10(0)` is undefined, so special handling is needed.

Final Notes

- This formula is efficient and avoids using loops.
- It works for **positive integers** but needs a check for `n = 0` or negative values.
- For **negative numbers**, you might take the absolute value before applying `log10`.