

Intro CP 6/2/26

loops ✓

- 1) HI Issues - intentionally planted
- 2) TLE, RTE

Integer Operations vs Floating Pts
LOSSY/
IMPRECISE

FLOOR function
CEILING function
ROUNDING

x/y using int division always
returns the floor (greatest int $\leq \frac{x}{y}$)
as long ~~pos~~ both are non-neg

$$17/3 \rightarrow 5$$

$$\lfloor \frac{x}{y} \rfloor = \text{floor, greatest int } \leq \frac{x}{y}$$

$$\lceil \frac{x}{y} \rceil = \text{ceiling, least int } \geq \frac{x}{y}$$

$$\text{res} = (x + y - 1) / y ; \quad \left\lceil \frac{21}{7} \right\rceil, \left\lceil \frac{22}{7} \right\rceil, \left\lceil \frac{28}{7} \right\rceil$$

3 4
→

If forced to use doubles

round $\rightarrow$ (int) $(x + .50000000001)$

$x + .5 + 1e-9$
 $\uparrow$
17
could be
spread as
16.9999...
16.4999...

int $[-2^{31}, 2^{31}-1]$

$\hookrightarrow \sim 2$ billion

OVERFLOW

$63212 * 2$
 71828

will be
wrong in C++!

long long $[-2^{63}, 2^{63}-1]$

$0 \leq a \leq 100,000$ } PAY ATTENTION!

Going Nuts

Given n , find the # of bits set to 1 in
 n 's binary representation.

$n = 13 \rightarrow$ 1101
 $2^3 2^2 2^1 2^0$

int div
mod

```
int res = 0;
while (n > 0) {
    res += n % 2;
    res = res /
    n = n / 2;
}
```

$n \% 2 \rightarrow$ lowest bit
least sig bit
 $n = n / 2 \rightarrow$ chops off
last bit

return res

BITWISE OPS

$$a = 1101 \quad (13)$$

$$b = 1011 \quad (11)$$

$$\underline{\underline{1001}}$$

$$13 \& 11 \rightarrow 9$$

$$a = 1101$$

$$b = 1011$$

$$\underline{\underline{1111}}$$

$$13 | 11 \rightarrow 15$$

$$a = 1101 \quad 13$$

$$\wedge b = 1011 \quad 11$$

$$\underline{\underline{0110}} \quad (6)$$

$$a \& b$$

↑

bitwise and

$$a | b$$

↑

bitwise OR

$$1010 \quad 10$$

$$1011 \quad 11$$

$$\underline{\underline{1011}}$$

$$10 | 11 \rightarrow 11$$

$$a \wedge b$$

↑

bitwise xor

(true if and only if exactly 1 bit is true)

$X \gg 1$ # bits to shift by

$X = 10101$, $X \gg 1 = 1010$

right shift by 1] $\div 2^k$

$X \ll 2$, left shift by 2] $\times 2^k$

$X = 10101$, $X \ll 2 = 1010100$

$X \& 1$ $\rightarrow$ $= 1$ if $lsb = 1$
 $= 0$ if $lsb = 0$

n Days Speed JP

Day 1 (1)

Day 2 $\underline{1} + (1+2) = (4)$

Day 3 $4 + (1+2+3) = (6)$

→ on day k I get $\frac{k(k+1)}{2}$ items.