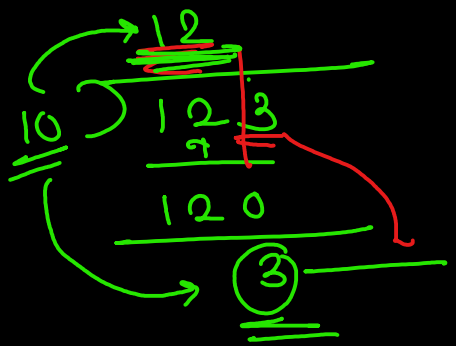
how to Find digit cnt in a number.

12345 | 6 ← 6 digit sum.

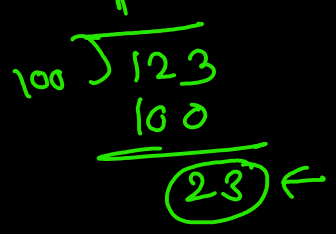
num ⇒ num/10 ⇒ Quotient → 12345.

Remainder ← last digit

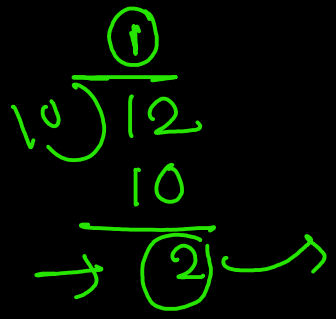


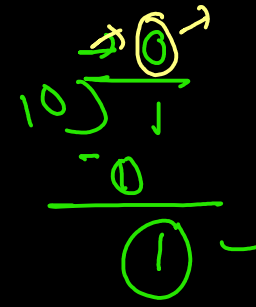
12 * 10 = 120

0 – 9



0 – 9





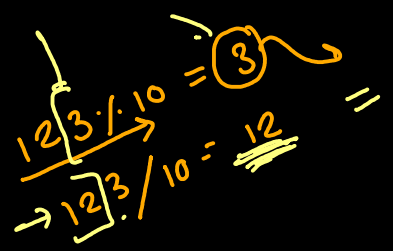
1 ← 2 ← 3

⟹ while (num > 0) {

↳ → last digit = num % 10 ←

↳ num = num / 10

}



→ Count digits

int cnt = 0 →

while (num > 0) {

cnt ++

num = num / 10 ←

}

num = ~~12345~~

↳ 1234

↳ ~~123~~

↳ ~~12~~

↳ 1

↳ 0 ✗

cnt = 0 + 1 + 1 + 1 + 1 ⑤

num = 123 → 12 → 1 → 0

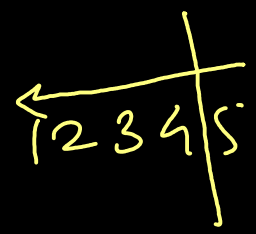
while (num > 0) {

cnt ++

num = num / 10

}

* . num % 2 = Remainder



prime or Not -

⇒ xe ← prime → Yes
→ NO.

Q. Full prime (GFG → https://www.geeksforgeeks.org/problems/full-prime2659/1?page=1&category=Prime%20Number&sortBy=submissions

⇒ num —
↳ 1st check the num is prime or not
↳ Yes → check it's digits are prime or not
↳ if all digits are prime
↳ full prime.
↳ else not full prime.
→ No it's not full prime

⇒ num 11, 12/

2 3 4 5 6 7 8 9 10

11

for (2 — num){

if (num % i == 0) return false.

}

num 37

2 — √num

2 3 4 5 6

√36 → 6 → 37

2 3 4 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19

20 21 22 23 24 25 26 27 28

29 30 31 32 33 34 35 36

36

2, 3, 6, 9, 12, 18/

for(i = 2 ; i*i <= n ; i++) {

if (n % i == 0) → return false.

}

n = 99 → 53

i = 2 =	i*1 = 9
i = 3,	i*1 = 9
i = 4	i*4 = 16
i = 5	i*1 = 25
i = 6	i*i = 36
i = 7	i*i = 49
i = 8	i*i = 64

1 2 3 4 5 6 7 = (7) → n = (√n)

O(√n) * dig(n)

O(√n) * log10(N)

0 ——— 9

2 3 5 7 ⇒

or 4 6 8 9 → Not prm