Basic Maths Concept :

N = 7789 % 10 = 9
/10 ↳ 778 % 10 = 8
↳ 77 % 10 = 7
/10 ↳ 7 % 10 = 7
/10 ↳ 0
/10

extraction of digits

```
while (n > 0) {
    last_digit = n % 10;
    n = n / 10;
}
```

2nd method: `int count = (int)(log10(n) + 1);`
`return count;`

Time complexity = $O(\log_{10}(N))$

if the division is happening n times → TC = $O(\log_{10}(N))$

GCD or HCF (Brute force) (linear complexity)

N1 = 9 N2 = 12 (gcd / hcf)(N, N2) = 3

(1)(3), 9 (1), 2, (3), 4, 6, 12

gcd (20/40) = 20

```
gcd = 1
for (i = 1 ; i <= min(n1,n2); i++){
    if (n1 % i == 0 && n2 % i == 0){
        gcd = i;
    }
}
```

// but we will use it ulta else faaltu me zyada baar ho sakta hai even ho jaaye.

```
for (i = min (n1, n2) ; i >= 1 ; i--){
    if (n1 % i == 0 && n2 % i == 0){
        print(i) ; break;
    }
}
```

// break hamesha loop hota hai, not if wali statement

TC : O(min(n1,n2))

GCD or HCF (Euclidean Algorithm)

(Better Approach)

$gcd(a,b) = gcd(a-b,b) \quad (a>b)$

$gcd(20,15) = gcd(5,15)$

$gcd(15,5) = gcd(10,5) = gcd(5,5) = gcd(0,5)$

$gcd = 5$

$gcd(a,b) \rightarrow gcd(a-b,b) \cdots \rightarrow$ one of them becomes 0

* but can become too long in some cases (eg: gcd(52,10))

$gcd(52,10) \rightarrow (42,10) \rightarrow (32,10) \rightarrow (22,10) \rightarrow (12,10) \rightarrow (2,10)$

$gcd(52\%10, 10)$

$gcd(10,2) \rightarrow (8,2) \rightarrow (6,2) \rightarrow (4,2) \rightarrow (2,2) \rightarrow (0,2)$

$gcd(10\%2, 2)$

$gcd(a, b) = gcd(a \% b, b)$ $\quad a > b$

if one of them is 0, the other is gcd.

```
while (a>0 && b>0){
    if (a>b) a=a%b;
    else     b=b%a;
}
if (a==0) cout<<b;
cout<<a;
```

$TC = O(\log_{\phi}(\min(a,b)))$ — $\min(a,b)$: initial num from where you started.

wherever division is happening : TC in terms of log.

ϕ : coz value of a & b keeps changing.