Trailing zeros In Factorial :

I/P : n = 5 $1 \times 2 \times 3 \times 4 \times 5 = 120$

O/P : 1

I/P : n = 10 $1 \times 2 \times 3 \times \cdots \times 9 \times 10$

$= 3628800$

O/P : 2

I/P : n = 100

O/P : 24

Mathematical logic :

First we must understand :

i) How are zeros made :

A trailing zero is produced by multiplying a number by 10.

Ex : $9 \times 10 = 90$ one trailing zero

ii) Prime factors :

Since $10 = 2 \times 5$

every pair of prime factors (2,5) in the factorial expansion adds one trailing zero.

Ex: 5!

$1 \times 2 \times 3 \times 4 \times 5$

one pair of (2, 5)

So 1 trailing zero

=> 120

↳ 1 trailing zero

10!

$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times \boxed{10}$

2×5

two pairs of (2, 5) So two trailing zeros

$10! = 3628800$

2 trailing zeros

Hence Logic is proved.

(ii) Limiting Factor:

In any factorial there are ~~plenty~~ of plenty of even numbers (multiples of 2) but multiples of 5 are much rarer.

Ex: In 20!

2, 4, 6, 8, 10, 12, 14, 16, 18, 20

i.e. multiples of 2

But only 2^3 multiples of 5 $(\overset{5}{10},20)$

Hence factors of 5 are bottleneck

We only need to count how many times 5 appears as a factor in the numbers from 1 to n. The number of 2s will always suffice to match them.

Trap:

We might think we just need to divide n by 5 to find the multiples of 5. 9×10

However some numbers contribute more than one 5

EX:

5, 10, 15, 20 → Each contribute one 5

$(1 \times 5,\ 2 \times 5,\ 3 \times 5,\ 4 \times 5)$

25 (5×5) → contribute 2 5's

125 $(5 \times 5 \times 5)$ → contribute 3 5's

So Count = $\left\lfloor \frac{n}{5} \right\rfloor + \left\lfloor \frac{n}{25} \right\rfloor + \left\lfloor \frac{n}{125} \right\rfloor + \dots$

Code :

```
int trailingZeros (int n)
{
    int count = 0;
    for (int i = 5 ; i <= n ; i = i * 5)
    {
        count = count + n/i ;
    }
    return count ;
}
```

Code Tracing :

Since in our logic to get count we use the formula

$$\left\lfloor \frac{n}{5} \right\rfloor + \left\lfloor \frac{n}{25} \right\rfloor + \left\lfloor \frac{n}{125} \right\rfloor + \dots$$

we will implement this formula using for loop

```
for (int i = 5; i <= n ; i = i * 5)
```

i) we will start initialization with 5

ii) Instead of increamenting by 1 we will increment by simply to

Reeeer...

next power of 5

i.e. $i = 5$ (First initialization)

$i = i * 5$

$(5 * 5) = 25$

$i = 25 * 5$

$= 125$

iii) Condition : $i <= n$

The loop continues as long as the power of 5 is less than or equal to n

EX : If $n = 14$ $\times 9 \times 10$

2

$i <= n$

$5 <= 14$ ✓

$25 <= 14$ ✗

So loop will run one time

If $n = 4$

$i <= n$

$5 <= 4$ ✗

Condition is not satisfied so loop will not be executed

In that case we have initialized count = 0 in the code

... count will remain as 0

Calculation part:

Trace the for loop

count = count + n/i

Initially count = 0

for first iteration i = 5

n/5 counts how many numbers contains atleast one factor of 5

for second iteration i = 25

n/25 counts how many numbers contains second factor of 5

And this goes on untill the condition i <= n is satisfied.

Ex 1:

n = 14

count = ~~0~~ 2

i = 5
n = 14

1st Iteration

5 <= 14 y

c = 0 + 14/5

c = 0 + 2
c = 2

second Iteration

5*5 <= 14 X

condition false

loop ends

count = 2

EX 2:

$n = 100$

Count $= \cancel{0}\ \cancel{20}\ 24$

$i = 5$

$n = 100$

1st

$5 <= 100$ ✓

$c = 0 + \frac{100}{5}$

$= 20$

2nd

$25 <= 100$ ✓

$c = 20 + \frac{100}{25}$

$= 24$

umbers

5

3rd

$125 <= 100$ ✗

Loop ends

numbers

Count $= 24$