

Count trailing zeroes in factorial of a number

Given an integer n , write a function that returns count of trailing zeroes in $n!$.

Examples:

Input: $n = 5$

Output: 1

Factorial of 5 is 20 which has one trailing 0.

Input: $n = 20$

Output: 4

Factorial of 20 is 2432902008176640000 which has 4 trailing zeroes.

Input: $n = 100$

Output: 24

A simple method is to first calculate factorial of n , then count trailing 0s in the result (We can count trailing 0s by repeatedly dividing the factorial by 10 till the remainder is 0).

The above method can cause overflow for a slightly bigger numbers as factorial of a number is a big number (See factorial of 20 given in above examples). The idea is to consider **prime factors** of a factorial n . A trailing zero is always produced by prime factors 2 and 5. If we can count the number of 5s and 2s, our task is done. Consider the following examples.

$n = 5$: There is one 5 and 3 2s in prime factors of 5! ($2 * 2 * 2 * 3 * 5$). So count of trailing 0s is 1.

$n = 11$: There are two 5s and three 2s in prime factors of 11! ($2^8 * 3^4 * 5^2 * 7$). So count of trailing 0s is 2.

We can easily observe that the number of 2s in prime factors is always more

than or equal to the number of 5s. So if we count 5s in prime factors, we are done. *How to count total number of 5s in prime factors of n!*? A simple way is to calculate $\text{floor}(n/5)$. For example, 7! has one 5, 10! has two 5s. It is done yet, there is one more thing to consider. Numbers like 25, 125, etc have more than one 5. For example if we consider 28!, we get one extra 5 and number of 0s become 6. Handling this is simple, first divide n by 5 and remove all single 5s, then divide by 25 to remove extra 5s and so on. Following is the summarized formula for counting trailing 0s.

$$\begin{aligned}\text{Trailing 0s in } n! &= \text{Count of 5s in prime factors of } n! \\ &= \text{floor}(n/5) + \text{floor}(n/25) + \text{floor}(n/125) + \dots\end{aligned}$$

Following is C++ program based on above formula.

```
// C++ program to count trailing 0s in n!
#include <iostream>
using namespace std;

// Function to return trailing 0s in factorial of n
int findTrailingZeros(int n)
{
    // Initialize result
    int count = 0;

    // Keep dividing n by powers of 5 and update count
    for (int i=5; n/i>=1; i *= 5)
        count += n/i;

    return count;
}

// Driver program to test above function
int main()
{
    int n = 100;
    cout << "Count of trailing 0s in " << 100
         << "! is " << findTrailingZeros(n);
    return 0;
}
```

Output:

Count of trailing 0s in 100! is 24