

Bài yêu cầu đếm chữ số 0 tận cùng của n!

Ta có : $n! = 1 \times 2 \times 3 \times 4 \times \dots \times n$

Mà ta lại có để tạo được số 0 ở cuối ta cần số 2 và số 5

Vậy ta cần đếm số lượng số 5 khi phân tích n! thành số nguyên tố (vì số lượng số 2 luôn lớn hơn số 5)

Ví dụ : $12! = 2^{10} \times 3^5 \times 5^2 \times 7 \times 11$ Có 2 số 5 nên sẽ tạo ra 2 số 0 ở cuối

Để đếm số lượng số 5 ta có thể làm như sau:

Các số chia hết cho 5 sẽ có dạng $5 \times x \rightarrow$ Các số chia hết cho 5 khi phân tích thừa số nguyên tố sẽ có ít nhất 1 số 5

Các số chia hết cho 5^2 sẽ có dạng $5^2 \times x \rightarrow$ Các số chia hết cho 5^2 khi phân tích thừa số nguyên tố có ít nhất 2 số 5

Các số chia hết cho 5^3 sẽ có dạng $5^3 \times x \rightarrow$ Các số chia hết cho 5^3 khi phân tích thừa số nguyên tố có ít nhất 3 số 5

.....

Vậy ta rút ra công thức sau :

Các số chia hết cho 5^m thì phân tích thừa số nguyên tố có ít nhất m số 5;

\rightarrow Để đếm số số 5 ta có thể dùng vòng lặp while và đếm xem có bao nhiêu số chia hết cho 5, 5^2 , 5^3 , ...

Code :

```
long long n ;cin >> n;
long long p = 5;// p lần lượt là các giá trị 5 , 5^2 , 5^3
long long cnt = 0;
while(p <= n){ // nếu p lớn hơn n -> số số chia hết cho p sẽ = 0 nên dừng chương trình
    cnt += n/p; // n/p là số lượng số chia hết cho p ?
    p*=5;// ví dụ 5 -> 5^2 , 5^2 -> 5^3
}
cout << cnt;
```

Trong đoạn code ta sẽ cộng những số chia hết cho 5, 25, 125,... lại vào với nhau Vì

Số chia hết cho 5 thì bao gồm cả số chia hết cho 25 ví dụ 25 là 1 số chia hết cho 5 nhưng 25 cũng chia hết cho 25

Mà số 25 đã đếm 1 lần tức là đã đếm 1 số 5 tách được từ số 25 , vậy nên số 25 hiện tại chỉ cung cấp 1 số 5

Tương tự các số 125 , ...