Problem Name : A Factorial Problem

Topic : Number Theory

Difficulty : Medium

EXPLANATION

A simple observation is that x will always be finite. Let us represent k as product of power of primes,

$k = p_1^{k_1} * p_2^{k_2} * \ldots * p_l^{k_l}$

Now let the occurrences of these primes, $p_1, p_2, \ldots, p_l$ in $n!$ be $c_1, c_2, \ldots, c_l$. If the number of occurrences of any of the primes $p_1, p_2, \ldots, p_l$ in k^x exceeds, $c_1, c_2, \ldots, c_l$ respectively, then $n!$ % k^x != 0. So we can conclude that, x is such that,

$k_1 * x <= c_1, k_2 * x <= c_2, \ldots, k_l * x <= c_l$

This means, $x = min(c_1/k_1, c_2/k_2, \ldots, c_l/k_l)$.

Time Complexity : O($sqrt(K)$)(Approx).

```
#include <bits/stdc++.h>
#define inf 1000000000
#define ll long long int
using namespace std;
int main()

{

    int t;

    cin >> t;

    while(t--)

    {

        int n, k;
```

```
        cin >> n >> k;

        int ans = inf;

        int mp = 1;

        for(int i=2; i*i<=k; i++) //prime factors less than s
qrt(k)

        {

            if(k % i == 0) // a prime factor is i.

            {

                mp = 0; // number of occurences of i in k.

                while(k % i == 0)

                {

                    mp++;

                    k /= i;

                }

                int tmp = 0;
```

```
            ll p = i;
            while(p <= n)
            {
                tmp += n / p;
                p*=i;
            }
            ans = min(ans, tmp / mp);
        }
    }
    if(k > 1) // prime factor greater than sqrt(k).
    {
        int tmp = 0;
        ll p = k;
        while(p <= n)
```

```
            {
                tmp += n / p;
                p *= k;
            }
            ans = min(ans, tmp);
        }
        if(ans == inf)
            ans = 0;
        cout << ans << endl;
    }
    return 0;
}
```