

Teaching Hazard

Teaching students is fun but it can often be embarrassing, which I experienced a few days ago. I was taking CSE3021 (Mathematical Analysis for Computer Science) class in my university and in the very first class I was teaching some very basic things. To be specific I was trying to teach students how to find trailing zeroes of $n!$ (factorial n) in base b . And of course many of you know that multiplicity of a prime factor p in $n!$ can be found using the formula

□

This formula can also be used cleverly to find number of trailing zeroes in $n!$.

After teaching this formula I showed them how to find number of trailing zeroes in $200!$ in decimal number system and with an evil smile asked them to find out number of trailing zeroes in $100!$ in hexadecimal (16-based) number system. I knew that the correct answer is 24 and to my utter surprise I got a correct reply from a student within minutes and so I congratulated him. But a minute later when I checked his script I found that he actually calculated number of trailing zeroes in $100!$ in decimal (not Hexadecimal) number system and coincidentally that both answers (Trailing zeroes in hexadecimal and decimal number system) were 24. So I was a bit embarrassed and now you have to help me find out why those two answers were same? Given a number n , you will have to find how many pair of bases (b_1, b_2) are there for which $n!$ (Factorial n) has exactly p trailing zeroes in both base b_1 and base b_2 . Here p is a positive integer not less than x .

Input

Input file contains 1000 lines of inputs. Each line contains two integers n ($1 \leq n \leq 100000$) and x ($2 \leq x \leq 2500$).

Input is terminated by a line containing two zeroes.

Output

For each line of input produce one line of output. This line contains an integer which denotes number of base pairs (b_1, b_2) so that $n!$ has exactly p trailing zeroes in both bases where p is not less than x . You can assume that inputs will be such that none of the output numbers will exceed $5 \cdot 10^{18}$.

Example

Input:

```
6 2
9 3
0 0
```

Output:

```
6
2
```