Homework #3

Due date: 10/28

Prime factorization and divisor

Given an unsigned integer $n \geq 2$, factor it into primes and use the factorization to determine the number and sum of divisors of n.

Let $n = {p_1}^{e_1}{p_2}^{e_2}\cdots{p_k}^{e_k}$, where $p_1 < p_2 < \cdots < p_k$ are primes and $e_i > 0$, be the prime factorization of n. Then,

$$\text{the number of divisors of } n = \prod_{i=1}^{k}(1+e_i) \qquad \cdots(1)$$

and

$$\text{the sum of divisors of } n = \prod_{i=1}^{k}\sum_{j=0}^{e_i} p_i^j \qquad \cdots(2)$$

For example, $20 = 2^2 \cdot 5$ has 6 divisors, namely, $1, 2, 4, 5, 10$ and 20, that sum up to 42, agreeing with formula (1): $(1+2)(1+1) = 6$ and formula (2): $(2^0 + 2^1 + 2^2)(5^0 + 5^1) = 42$.

Requirements

1 You shall write a function, say

`void factorization(unsigned n);` // use unsigned type

to factor n and compute the number and sum of its divisors. As an example, the call **`factorization(20)`** shall output

```
Prime factorization of 20 = 2^2x5^1
Number of divisors = 6
Sum of divisors = 42
```

2 The easiest factorization method is the trial division algorithm that consists of the following loop:

while (*not finish yet*) {

Find the next prime p

Find the largest integer e such that p^e divides n

Reduce n to n/p^e

}

For example, let $n = 20 = 2^2 \cdot 5$, the values of $p, e,$ and n at the end of each iteration are shown below:

1st iteration	$p = 2$	$e = 2$	$n = 5$
2nd iteration	$p = 3$	$e = 0$	$n = 5$
3rd iteration	$p = 5$	$e = 1$	$n = 1$

In this case, the loop terminates when $n = 1$. In other cases, we don't need to wait until $n = 1$ to terminate the loop.
For example, let $n = 84 = 2^2 \cdot 3 \cdot 7$, then

1st iteration	$p = 2$	$e = 2$	$n = 21$	
2nd iteration	$p = 3$	$e = 1$	$n = 7$	
3rd iteration	$p = 5$	$e = 0$	$n = 7$	(redundant)
4th iteration	$p = 7$	$e = 1$	$n = 1$	(redundant)

The last two iterations are redundant, because n is already a prime at the end of the 2nd iteration.
Figure out a good termination condition for the loop.
Be careful of integer overflow.

3 You shall compute the value of p_i^j incrementally. That is, do not compute p_i^j from scratch. Instead, use the value of p_i^{j-1} to compute p_i^j.

4 Refer to the sample run below for the required output format.

Sample run

```
Enter an unsigned integer >= 2: 20
Prime factorization of 20 = 2^2x5^1
Number of divisors = 6
Sum of divisors = 42

Enter an unsigned integer >= 2: 84
Prime factorization of 84 = 2^2x3^1x7^1
Number of divisors = 12
Sum of divisors = 224
```

```
Enter an unsigned integer >= 2: 3287037600
Prime factorization of 3287037600 = 2^5x3^2x5^2x7^3x11^3
Number of divisors = 864
Sum of divisors = 1982896512

Enter an unsigned integer >= 2: 4198216889
Prime factorization of 4198216889 = 60917^1x68917^1
Number of divisors = 4
Sum of divisors = 4198346724

Enter an unsigned integer >= 2: 4294967279
Prime factorization of 4294967279 = 4294967279^1
Number of divisors = 2
Sum of divisors = 4294967280

Enter an unsigned integer >= 2: 4294967295
Prime factorization of 4294967295 = 3^1x5^1x17^1x257^1x65537^1
Number of divisors = 32
Sum of divisors = 3009636032

Enter an unsigned integer >= 2: ^Z
```