Counting Divisors (general)

Let $\sigma_0(n)$ be the number of positive divisors of n.

For example, $\sigma_0(1) = 1$, $\sigma_0(2) = 2$ and $\sigma_0(6) = 4$.

Let $$S_k(n) = \sum _{i=1}^n \sigma_0(i^k).$$

Given n and k, find $S_k(n) \bmod 2^{64}$.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 10000$), indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n, k \le 10^{10}$).

Output

For each test case, output a single line containing $S_k(n) \bmod 2^{64}$.

Example

Input

```
5
1 3
2 3
3 3
10 3
100 3
```

Output

```
1
5
9
73
2302
```

Information

There are 5 Input files.

- Input #1: $1 \le n \le 10000$, TL = 1s.
- Input #2: $1 \le T \le 300,\ 1 \le n \le 10^7$, TL = 5s.
- Input #3: $1 \le T \le 75,\ 1 \le n \le 10^{8}$, TL = 5s.
- Input #4: $1 \le T \le 15,\ 1 \le n \le 10^{9}$, TL = 5s.
- Input #5: $1 \le T \le 5,\ 1 \le n \le 10^{10}$, TL = 5s.

My C++ solution runs in 5.6 sec. (total time)

Notes

This is general version of DIVCNT1, DIVCNT2 and DIVCNT3. You may want to solve these three problems first.