Problem E

Inverse Divisor Sums

Your friend Odd Even is obsessed with number theory. He likes to learn new operations to perform on numbers, and to spend endless hours applying his newly acquired knowledge to numbers. For instance, last year he learned about Euler's totient function $\phi(n)$, which counts the number of positive integers less than or equal to n that are relatively prime to n. Shortly after that, he went on a spree and calculated $\phi(n)$ for all integers n from 1 up to one million by hand.

Recently, he learned that the sum of all divisors of a number N could be calculated by the following formula:

$$\text{sum of divisors}(N) = \prod_{i=1}^{r} \frac{p_i^{(a_i+1)} - 1}{p_i - 1}.$$

Here, $p_1^{a_i} p_2^{a_2} \ldots p_r^{a_r}$ is the factorization of N into prime factors where each p_i is different and a_i is the maximum power of p_i such that $p_i^{a_i}$ that divides N.

Odd Even wants to calculate the function in reverse; given a positive integer N he wants to find all positive integers M having N as its sum of divisors, and he wants them written out nicely in increasing order. You think this will take too long, so you have decided to intervene and offer computer assistance.

Write a computer program that does the following: given a positive integer N, output a list of all the integers M having N as its sum of divisors, in increasing order, or inform Odd Even that no such numbers exists.
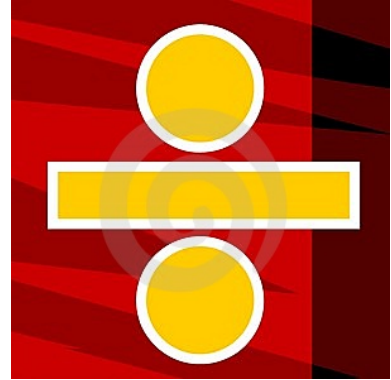
Input specifications

The first line of the input consists of a single integer T, the number of test cases. The T following lines each contain a single integer N.

Output specifications

For each test case, output all such numbers on one line, in increasing order, with a single space between each number. If no numbers exist, output "`none!`" (without quotes).

Notes and Constraints

- $0 < N \leq 10^9$
- The output can be large. For Java, it is advisable to buffer the output using StringBuilder.

Sample input	Output for sample input
4	
7	4
2	none!
126	68 82
1524	704 1083 1523