

Problem D. More Divisors

Input file: `divisors.in`
Output file: `divisors.out`
Time limit: 1 second
Memory limit: 64 megabytes

Everybody knows that we use decimal notation, i.e. the base of our notation is 10. Historians say that it is so because men have ten fingers. Maybe they are right. However, this is often not very convenient, ten has only four divisors — 1, 2, 5 and 10. Thus, fractions like $1/3$, $1/4$ or $1/6$ have inconvenient decimal representation. In this sense the notation with base 12, 24, or even 60 would be much more convenient.

The main reason for it is that the number of divisors of these numbers is much greater — 6, 8 and 12 respectively. A good question is: what is the number not exceeding n that has the greatest possible number of divisors? This is the question you have to answer.

Input

The input file n ($1 \leq n \leq 10^{16}$).

Output

Output positive integer number that does not exceed n and has the greatest possible number of divisors. If there are several such numbers, output the smallest one.

Example

| <code>divisors.in</code> | <code>divisors.out</code> |
|--------------------------|---------------------------|
| 10 | 6 |
| 20 | 12 |
| 100 | 60 |