**Prime Factorization of a Number**

Given a positive integer N, your task is to find its prime factorization. Return a list of prime numbers that multiply together to give N. If N is prime, the output should be a list containing only N.

Prime factorization is the process of breaking down a number into the set of prime numbers that, when multiplied together, result in the original number. For example, if N = 18, its prime factors are [2, 3, 3] because 2×3×3=18.

**Input:**

* A single integer N, where 2≤N≤109

**Output:**

* A list of prime numbers representing the prime factorization of N.

**Examples:**

* Example 1  
  Input: N = 18

Output: [2, 3, 3]  
Explanation: The prime factorization of 18 is 2 × 3 × 3.

**Constraints:**

* 2≤N≤109
* N is guaranteed to be a positive integer.

**Test Cases:**

1. Input: N = 30

Output: [2, 3, 5]

1. Input: N = 49

Output: [7, 7]

1. Input: N = 19

Output: [19]

1. Input: N = 64

Output: [2, 2, 2, 2, 2, 2]

1. Input: N = 123456

Output: [2, 2, 2, 2, 2, 3, 643]

**Edge Cases:**

1. N is a small prime number: If N is prime (e.g., 2, 3, 5, 7), the list should only contain N.
2. N is a perfect power of a prime: For N = 27, the result should be [3, 3, 3].
3. N has large prime factors: Ensure the algorithm can handle prime factors larger than the square root of N.