Naïve Factorization

In this problem, we are interested in what the factorization of an integer N is. For example, $200 = 2^3 \times 5^2$ or $3298402 = 2 \times 29^2 \times 37 \times 53$ Write the function factor(N) that returns a list [[p1, n1], [p2, n2], ..., [pk,nk]] where $N = \prod_{i=1}^{k} p_i^{n_i}$. For example:

- factor(200) returns [[2,3], [5,2]]
- factor(3298402) returns [[2,1], [29,2], [37,1], [53,1]]

```
def factor(N): #N is a positive integer greater than 1


exec(input().strip()) #This line is required for grader to work.
```

A simple method of finding all factors of an integer N can be done by brute forcing dividing numbers from k = 2, 3, ..., and so on. Whenever a value of N is divisible by k with no remainders, continuously divide N with k until it cannot be divided anymore (while recording how many times this division is done). Then, try the next k value. Doing this will increase k and decrease N. Whenever k is greater than N, this means all the factors have been found.

Example: Let N = 200. Start at k = 2. 200 is divisible by 2. N can be divided 4 times until it can no longer be divided. N changes to 200 → 100 → 50 → 25 (Add [2,3] to the answer). Then, change to k=3. 25 isn't divisible by 3, so this is skipped. 4 is also skipped. When k=5, it is divisible twice, changing N to 25 → 5 → 1. (Add [5,2] to the answer). Now, k=6 and N=1. All factors are now found, the answer is [[2,3],[5,2]], which represents $200 = 2^3 \times 5^2$.

Input

A Python command to test the function.

Output

The result after running the command.

Example

Input (from keyboard)	Output (on screen)
`print(factor(200))`	`[[2, 3], [5, 2]]`
`print(factor(3298402))`	`[[2, 1], [29, 2], [37, 1], [53, 1]]`
`print(factor(8137740897))`	`[[3, 4], [11, 2], [13, 2], [17, 3]]`