Making Candies

Karl loves playing games on social networking sites. His current favorite is *CandyMaker*, where the goal is to make candies.

Karl just started a level in which he must accumulate n candies starting with m machines and w workers. In a single *pass*, he can make $m \times w$ candies. After each pass, he can decide whether to spend some of his candies to buy more machines or hire more workers. Buying a machine or hiring a worker costs p units, and there is no limit to the number of machines he can own or workers he can employ.

Karl wants to minimize the number of passes to obtain the required number of candies at the end of a day. Determine that number of passes.

For example, Karl starts with $m = 1$ machine and $w = 2$ workers. The cost to purchase or hire, $p = 1$ and he needs to accumulate 60 candies. He executes the following strategy:

1. Make $m \times w = 1 \times 2 = 2$ candies. Purchase two machines.
2. Make $3 \times 2 = 6$ candies. Purchase 3 machines and hire 3 workers.
3. Make $6 \times 5 = 30$ candies. Retain all 30 candies.
4. Make $6 \times 5 = 30$ candies. With yesterday's production, Karl has 60 candies.

It took 4 passes to make enough candies.

Function Description

Complete the *minimumPasses* function in the editor below. The function must return a long integer representing the minimum number of passes required.

minimumPasses has the following parameter(s):

- *m*: long integer, the starting number of machines
- *w*: long integer, the starting number of workers
- *p*: long integer, the cost of a new hire or a new machine
- *n*: long integer, the number of candies to produce

Input Format

A single line consisting of four space-separated integers describing the values of m, w, p, and n, the starting number of machines and workers, the cost of a new machine or a new hire, and the the number of candies Karl must accumulate to complete the level.

Constraints

- $1 \leq m, w, p, n \leq 10^{12}$

Output Format

Return a long integer denoting the minimum number of passes required to accumulate at least n candies.

Sample Input

```
3 1 2 12
```

Sample Output

```
3
```

Explanation

Karl makes three passes:

1. In the first pass, he makes $m \times w = 3 \times 1 = 3$ candies. He then spends $p = 2$ of them hiring another worker, so $w = 2$ and he has one candy left over.
2. In the second pass, he makes $3 \times 2 = 6$ candies. He spends $2 \cdot p = 4$ of them on another machine and another worker, so $w = 3$ and $m = 4$ and he has 3 candies left over.
3. In the third pass, Karl makes $4 \times 3 = 12$ candies. Because this satisfies his goal of making at least $n = 12$ candies, we print the number of passes (i.e., 3) as our answer.