Making Candies

by Live_Forever

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 make n candies using 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 of candies, and there is no limit to the number of machines he can build or workers he can hire.

Karl wants to maximize his score by making all n candies in a minimum number of passes. Can you find and print the minimum number of passes required for Karl to make at least n units of candies?

Input Format

A single line consisting of four space-separated integers describing the respective values of m (the number of machines), w (the number of workers), p (the price of buying one machine or hiring one worker), and n (the number of candies Karl must make).

Constraints

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

Output Format

Print the minimum number of passes required to make 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.

Difficulty: Hard

Rate This Challenge:

More

Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    static long minimumPasses(long m, long w, long p, long n) {
        // Complete this function
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        long m = in.nextLong();
        long w = in.nextLong();
        long p = in.nextLong();
        long n = in.nextLong();
        long result = minimumPasses(m, w, p, n);
        System.out.println(result);
        in.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code