3 Questions | Total Marks: 300.0

2 Programming Questions

1. Gift Vouchers + 100.0

2. Food Orders + 100.0

1 Approximate Question

3. Online Shopping + 100.0

Question 2 | Max. Marks 100.00

Food Orders

The company X supplies food products in the market. Since the demand is very large, they came with an idea of making a large warehouse with a capacity of N items. Since food products can't be kept for a long time, they started to expire. The dates of expiry are such that every day exactly K of the N food items get expired. Company X has n vans and each van can deliver m products in a day. How many minimum additional vans should the company X buy so that not more than p products get expire before they get delivered.

Note: Everyday the vans go and deliver the product after that K items expire.

Input

The first line contains 5 space-separated integers as input: N, K, n, m, and p respectively.

Output

In the output, you need to print the minimum number of extra vans required.

Constraints

$1 \leq N, K, n, m, p \leq 10^{18}$

Sample Input	Sample Output
10 1 3 1 1	2

Explanation

Let us assume that the company does not buy any van. So, on the first day, 1 item will expire and 3 gets delivered. On the second day, 1 more item gets expired and 3 more gets delivered. On the third day, the remaining 2 items get delivered.

So, out of the 10 items, 8 get successfully delivered and 2 expire. Since expired items are greater than 1, we need to increase the number of vans.

If we add just one more van, you see that 4 items get delivered on the first day and one item gets expired. 4 items again get delivered and 1 more item gets expired. Thus out of the 10 items, 8 gets delivered and 2 gets expired. Still, expired items are more than 1.

Now, we add 2 more vans so 5 items get delivered and 1 expires on the first day. Now, remaining 4 items get delivered. So only 1 item expires.

Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge. because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Marks are awarded if any testcase passes
Allowed Languages: Java, Java 8

New Submission | All Submissions

Java (openjdk 1.7.0_95) ▾ Save

```
/* IMPORTANT: Multiple classes and nested static classes are supported */

/*
 * uncomment this if you want to read input.
//imports for BufferedReader
import java.io.BufferedReader;
import java.io.InputStreamReader;

//import for Scanner and other utility classes
import java.util.*;
*/

// Warning: Printing unwanted or ill-formatted data to output will cause the test cases to fail

class TestClass {
    public static void main(String args[] ) throws Exception {
        /* Sample code to perform I/O:
         * Use either of these methods for input

        //BufferedReader
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String name = br.readLine();                // Reading input from STDIN
        System.out.println("Hi, " + name + ".");    // Writing output to STDOUT

        //Scanner
        Scanner s = new Scanner(System.in);
        String name = s.nextLine();                 // Reading input from STDIN
        System.out.println("Hi, " + name + ".");    // Writing output to STDOUT

        */

        // Write your code here

    }
}
```

1:1

Press Ctrl/Command+Spacebar for autocomplete suggestions (accuracy dependent on connection stability).

Provide custom input

COMPILE & TEST | SUBMIT

Next Question ›