Problem A. Quantum Supremacy

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

The era of quantum computers is just around the corner. At least, that is promised by different reports almost every year. Quantum computers are very expensive to build and maintain but can result in orders of magnitude speed-up in some specific tasks. The key parameter of a quantum computer is the number of qubits q it operates with.

Suppose there is a specific cryptographic algorithm that uses a secret binary string of length n. The only known way to crack it using a traditional computer is to try all 2^n possible secret strings and test each in a seconds. For a quantum computer with q qubits there is an algorithm that can simultaneously test up to 2^q secret binary strings in b seconds. These can be arbitrary $k \le 2^q$ binary strings of length n, chosen among all 2^n binary strings of this length.

You want to know when the quantum supremacy starts. Given n, a and b, determine the minimum non-negative integer value of q such that the time it takes to solve the problem (test all secret strings) using a quantum computer only is strictly less than the time it takes to solve the problem using a traditional computer.

Input

The only line of the input contains three integers n, a and b ($1 \le n, a, b \le 10^{18}$), the length of the secret binary string used by cryptography algorithm, the time it takes to test one binary string with a traditional computer and the time it takes to test up to 2^q binary strings using a quantum computer.

Output

If there is no non-negative intever value of q that results in quantum supremacy for this problem, print `-1` in the only line of the output.

Otherwise, print the minimum non-negative integer q such that a quantum computer with q qubits will be faster than a traditional computer.

Examples

standard input	standard output
1024 1 1	1
1024 1 2	2