Bomb, Baby!

You're so close to destroying the LAMBCHOP doomsday device you can taste it! But in order to do so, you need to deploy special self-replicating bombs designed for you by the brightest scientists on Bunny Planet. There are two types: Mach bombs (M) and Facula bombs (F). The bombs, once released into the LAMBCHOP's inner workings, will automatically deploy to all the strategic points you've identified and destroy them at the same time.

But there's a few catches. First, the bombs self-replicate via one of two distinct processes:
Every Mach bomb retrieves a sync unit from a Facula bomb; for every Mach bomb, a Facula bomb is created;
Every Facula bomb spontaneously creates a Mach bomb.

For example, if you had 3 Mach bombs and 2 Facula bombs, they could either produce 3 Mach bombs and 5 Facula bombs, or 5 Mach bombs and 2 Facula bombs. The replication process can be changed each cycle.

Second, you need to ensure that you have exactly the right number of Mach and Facula bombs to destroy the LAMBCHOP device. Too few, and the device might survive. Too many, and you might overload the mass capacitors and create a singularity at the heart of the space station - not good!

And finally, you were only able to smuggle one of each type of bomb - one Mach, one Facula - aboard the ship when you arrived, so that's all you have to start with. (Thus it may be impossible to deploy the bombs to destroy the LAMBCHOP, but that's not going to stop you from trying!)

You need to know how many replication cycles (generations) it will take to generate the correct amount of bombs to destroy the LAMBCHOP. Write a function solution(M, F) where M and F are the number of Mach and Facula bombs needed. Return the fewest number of generations (as a string) that need to pass before you'll have the exact number of bombs necessary to destroy the LAMBCHOP, or the string "impossible" if this can't be done! M and F will be string representations of positive integers no larger than 10^50. For example, if M = "2" and F = "1", one generation would need to pass, so the solution would be "1". However, if M = "2" and F = "4", it would not be possible.

Languages

To provide a Java solution, edit Solution.java
To provide a Python solution, edit solution.py

Test cases
==========
Your code should pass the following test cases.
Note that it may also be run against hidden test cases not shown here.

-- Java cases --
Input:
Solution.solution('2', '1')
Output:
    1

Input:
Solution.solution('4', '7')
Output:
    4

-- Python cases --
Input:
solution.solution('4', '7')
Output:
    4

Input:
solution.solution('2', '1')
Output:
    1

Use verify [file] to test your solution and see how it does. When you are finished editing your code, use submit [file] to submit your answer. If your solution passes the test cases, it will be removed from your home folder.