

1. Logic Building Task

Cancer Cell Treatment using Radioactive Materials

You are a scientist studying the effects of two types of radioactive materials, Mach (M) and Facula (F), on the human body. The materials, once introduced into the body, will automatically replicate themselves and affect specific cells.

But there are a few catches. First, the materials self-replicate via one of two distinct processes:

Every Mach material interacts with a Facula material; for every Mach material, a Facula material is created;

Every Facula material spontaneously creates a Mach material.

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

Second, you need to ensure that you have exactly the right number of Mach and Facula materials to affect the targeted cells. Too few, and the cells might be unaffected. Too many, and you might cause serious damage to the body - not good!

And finally, you were only able to introduce one of each type of material - one Mach, one Facula - into the body, so that's all you have to start with. (Thus it may be impossible to affect the targeted cells, 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 materials to affect the targeted cells. Write a function `solution(M, F)` where M and F are the number of Mach and Facula materials needed. Return the fewest number of generations (as a string) that need to pass before you'll have the exact number of materials necessary to affect the targeted cells, 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.

Example 1

Input: `solution('4', '7')`

Output: 4

Example 2

Input: `solution('2', '1')`

Output: 1

2. Data Science Task

Shape Detection

You are provided with a dataset containing 35 cropped images of traffic signs where each image contains one traffic sign. There are 7 different categories of shapes with 5 images per category. Please refer to Figure 1. Your task is to develop a generic algorithm which takes an image and outputs the shape of the traffic sign detected in the image. The images can be found in the attached .zip file named “task2_dataset.zip”. The code should be written in Python.

Sample Input



Sample Output

Circle

Shape	Class	Image
Circle	Do No Enter	
Square	Speed Limit	


Shape	Class	Image
Horizontal Rectangle	Information	
Vertical Rectangle	Object Marker	
Diamond	Left Turn	
Octagon	Stop	
Pentagon	School	

Figure 1. Table showing the shape, class, and sample image of traffic signs.