Jack isn't really good at math, but his sadistic teacher still gave him some math exercises about sequences. He needs to find the next number of the given sequence, which can be one of the following three types:

* arithmetic progression;
* geometric progression;
* pseudo-fibonacci sequence (a sequence each element of which is the sum of two preceding elements).

You are given 4 first numbers in the series as a string, in which the numbers are separated by commas. Help Jack to figure out what type of the sequence it is, and return the next number.

**Example**

* For sequence = "4,6,10,16", the output should be  
  CompletetheSequence(sequence) = 26.

The sequence is a pseudo-fibonacci sequence.

* For sequence = "5,10,20,40", the output should be  
  CompletetheSequence(sequence) = 80.

The sequence for a geometric progression with r = 2.

* For sequence = "2,4,6,8", the output should be  
  CompletetheSequence(sequence) = 10.

The sequence for a arithmetic progression with d = 2.

**Input/Output**

* **[time limit] 4000ms (py)**
* **[input] string sequence**

A string containing 4 numbers separated by a comma.

*Constraints:*  
7 ≤ sequence.length ≤ 254.

* **[output] integer**

The next element of the sequence.

<https://codefights.com/challenge/QLdW6grw8hEr75tLu?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

def **CompletetheSequence**(sequence):

s = sequence.split(*","*)

if int(s[0]) + int(s[1]) == int(s[2]) and int(s[1]) + int(s[2]) == int(s[3]):

return int(s[2]) + int(s[3])

arit = int(s[1]) - int(s[0])

if int(s[1]) + arit == int(s[2]):

return int(s[3]) + arit

dif = []

for i in range(0, len(s) - 1):

dif.append(int(s[i+1]) - int(s[i]))

if int(dif[1]) - int(dif[0]) == int(dif[2]) - int(dif[1]):

return int(s[3]) + dif[2] + int(dif[2]) - int(dif[1])

if int(s[1]) / int(s[0]) == int(s[2])/int(s[1]):

return int(s[3]) \* int(s[2])/int(s[1])