*"""  
Write python code for a recursive algorithm that will calculate the number of digits in the  
binary expansion/representation of a positive integer n. The logic of the recursive algorithm  
should be something like:  
  
if n = 1, the answer is 1;  
if n > 1, the answer is 1 more than the number of digits in the binary representation of  
n/2.  
"""*import math  
  
  
def binary\_digits(n):  
 *"""  
 Determines the number of digits in the binary expansion of a positive integer* ***:param*** *n: positive integer value* ***:return*** *digits: number of digits in binary expansion  
 """* if n == **1**:  
 digits = **1** return digits  
 elif n > **1**:  
 digits = **1** + binary\_digits(math.floor(n / **2**))  
 return digits  
  
  
"""  
Write python code for a recursive algorithm that will calculate the sum of the squares of the  
positive integers 12 + 22 + 32 + … + 𝑛𝑛2 when supplied with a positive integer n.  
The logic of the recursive algorithm should be something like:  
if n = 1, the answer is 1;  
if n > 1, the answer is (the sum of the squares of the integers from 1 to n-1) + 𝑛2.   
"""  
  
  
def sum\_of\_squares(n):  
 *"""  
 Calculates the sum of squares of consecutive integers from 1 to n* ***:param*** *n: stopping point integer* ***:return****: sum of squares  
 """* if n == **1**:  
 return **1** if n > **1**:  
 return n\*\***2** + sum\_of\_squares(n-**1**)  
  
  
def main():  
 # Answers to Question 1b  
 x = binary\_digits(**256**)  
 y = binary\_digits(**750**)  
  
 print("Answers to Question 1b: ")  
 print(x**,** y)  
  
 # Answers to Question 2b  
 a = sum\_of\_squares(**12**)  
 b = sum\_of\_squares(**20**)  
  
 print("Answers to Question 2b: ")  
 print(a**,** b)  
  
  
main()

Answers to Question 1b:

9 10

Answers to Question 2b:

650 2870

A picture containing text, handwriting, letter, ink

Description automatically generated