**Question1. Create a function that takes three arguments a, b, c and returns the sum of the**

**numbers that are evenly divided by c from the range a, b inclusive.**

**Examples**

**evenly\_divisible(1, 10, 20) ➞ 0**

**# No number between 1 and 10 can be evenly divided by 20.**

**evenly\_divisible(1, 10, 2) ➞ 30**

**# 2 + 4 + 6 + 8 + 10 = 30**

**evenly\_divisible(1, 10, 3) ➞ 18**

**# 3 + 6 + 9 = 18**

def evenly\_divisible(a, b, c):

result = 0

for i in range(a, b+1):

if i % c == 0:

result += i

return result

print(evenly\_divisible(1, 10, 20)) # 0

print(evenly\_divisible(1, 10, 2)) # 30

print(evenly\_divisible(1, 10, 3)) # 18

**Question2. Create a function that returns True if a given inequality expression is correct and**

**False otherwise.**

**Examples**

**correct\_signs(&quot;3 &lt; 7 &lt; 11&quot;) ➞ True**

**correct\_signs(&quot;13 &gt; 44 &gt; 33 &gt; 1&quot;) ➞ False**

**correct\_signs(&quot;1 &lt; 2 &lt; 6 &lt; 9 &gt; 3&quot;) ➞ True**

def correct\_signs(inequality):

elements = inequality.split()

for i in range(len(elements) - 2):

if elements[i + 1] == "<":

if not (int(elements[i]) < int(elements[i + 2])):

return False

elif elements[i + 1] == ">":

if not (int(elements[i]) > int(elements[i + 2])):

return False

return True

print(correct\_signs("3 < 7 < 11")) # True

print(correct\_signs("13 > 44 > 33 > 1")) # False

print(correct\_signs("1 < 2 < 6 < 9 > 3")) # True

**Question3. Create a function that replaces all the vowels in a string with a specified character.**

**Examples**

**replace\_vowels(&quot;the aardvark&quot;, &quot;#&quot;) ➞ &quot;th# ##rdv#rk&quot;**

**replace\_vowels(&quot;minnie mouse&quot;, &quot;?&quot;) ➞ &quot;m?nn?? m??s?&quot;**

**replace\_vowels(&quot;shakespeare&quot;, &quot;\*&quot;) ➞ &quot;sh\*k\*sp\*\*r\*&quot;**

def replace\_vowels(string, character):

vowels = "aeiouAEIOU"

result = ""

for char in string:

if char in vowels:

result += character

else:

result += char

return result

print(replace\_vowels("the aardvark", "#")) # "th# ##rdv#rk"

print(replace\_vowels("minnie mouse", "?")) # "m?nn?? m??s?"

print(replace\_vowels("shakespeare", "\*")) # "sh\*k\*sp\*\*r\*"

**Question4. Write a function that calculates the factorial of a number recursively.**

**Examples**

**factorial(5) ➞ 120**

**factorial(3) ➞ 6**

**factorial(1) ➞ 1**

**factorial(0) ➞ 1**

def factorial(n):

if n == 0:

return 1

elif n == 1:

return 1

else:

return n \* factorial(n - 1)

print(factorial(5)) # 120

print(factorial(3)) # 6

print(factorial(1)) # 1

print(factorial(0)) # 1

**Question 5**

**Hamming distance is the number of characters that differ between two strings.**

**To illustrate:**

**String1: &quot;abcbba&quot;**

**String2: &quot;abcbda&quot;**

**Hamming Distance: 1 - &quot;b&quot; vs. &quot;d&quot; is the only difference.**

**Create a function that computes the hamming distance between two strings.**

**Examples**

**hamming\_distance(&quot;abcde&quot;, &quot;bcdef&quot;) ➞ 5**

**hamming\_distance(&quot;abcde&quot;, &quot;abcde&quot;) ➞ 0**

**hamming\_distance(&quot;strong&quot;, &quot;strung&quot;) ➞ 1**

def hamming\_distance(string1, string2):

distance = 0

for i in range(len(string1)):

if string1[i] != string2[i]:

distance += 1

return distance

print(hamming\_distance("abcde", "bcdef")) # 5

print(hamming\_distance("abcde", "abcde")) # 0

print(hamming\_distance("strong", "strung")) # 1