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

A class div():

def \_\_init\_\_(self , a,b,c):

self.a=a

self.b=b

self.c=c

def evenly\_divisible(self):

sum=0

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

if(i%self.c==0):

sum+=i

return sum

a=int(input())

b=int(input())

c=int(input())

sum1=div(a,b,c)

print(sum1.evenly\_divisible())print(evenly\_divisible(a,b,c))

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

A:

def correct\_signs ( string ) :

return eval ( string )

print(correct\_signs("3 > 7 < 11"))

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

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

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;

A:

minnie mouse

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

A:

def factorial(n):

# Checking the number

# is 1 or 0 then

# return 1

# other wise return

# factorial

if (n==1 or n==0):

return 1

else:

return (n \* factorial(n - 1))

# Driver Code

num = 5;

print("number : ",num)

print("Factorial : ",factorial(num))

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

A:

def hamming\_distance(str1, str2):

i = 0

count = 0

while(i < len(str1)):

if(str1[i] != str2[i]):

count += 1

i += 1

return count

# Driver code

str1 = "abcde"

str2 = "bcdef"

# function call

print(hamming\_distance(str1, str2))