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

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

### Examples

correct\_signs("3 < 7 < 11") ➞ True

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

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("the aardvark", "#") ➞ "th# ##rdv#rk"

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

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

**Question 5**

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

To illustrate:

String1: "abcbba"

String2: "abcbda"

Hamming Distance: 1 - "b" vs. "d" is the only difference.

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

### Examples

hamming\_distance("abcde", "bcdef") ➞ 5

hamming\_distance("abcde", "abcde") ➞ 0

hamming\_distance("strong", "strung") ➞ 1

**Solution: 1**

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

sum = 0

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

if (i%c==0):

sum= sum+i

return sum

evenly\_divisible(1,10,3)

**Solution: 2**

def correct\_signs(s):

if(s):

return(True)

else:

return(False)

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

**Solution: 3**

text = "Amar"

vowels = 'AEIOUaeiou'

for i in vowels:

text = text.replace(i, '#')

print(text)

**Solution: 4**

def fact(n):

if(n==1):

return 1

else:

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

n = int(input("Enter a number "))

fact(n)

**Solution: 5**

def hamming\_distance(text1, text2):

count = 0

for i in range(0, len(text1)):

if(text1[i]!=text2[i]):

count = count+1

return count

hamming\_distance("geekspractice", "nerdspractise")