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 sum\_num(a, b, c):

sum = 0

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

if i % c == 0:

sum += i

return sum

a = int(input("Enter first number : "))

b = int(input("Enter second number : "))

c = int(input("Enter third number : "))

print(sum\_num(a, b, c))

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

def checkEquality():

in\_string = input('Enter the inequality: ')

out\_bool = eval(in\_string)

print(f'{in\_string} ➞ {out\_bool}')

for x in range(4):

checkEquality()

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\*"

def replaceVowels():

vowels = ['a','e','i','o','u','A','E','I','O','U']

in\_string = input("String: ")

in\_string\_copy = in\_string

in\_char = input('Replacement character: ')

new\_chars = [in\_char if char in vowels else char for char in in\_string]

modified\_string = ''.join(new\_chars)

print(f'{in\_string\_copy} {in\_char} ➞ {modified\_string}')

for x in range(3):

replaceVowels()

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

else:

return n \* factorial(n-1)

num = int(input("Enter number : "))

print(factorial(num))

**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

def hamming\_distance(s1, s2):

if len(s1) != len(s2):

raise ValueError("Both strings must have the same length")

distance = 0

for c1, c2 in zip(s1, s2):

if c1 != c2:

distance += 1

return distance

s1 = input("Enter first string : ")

s2 = input("Enter second string : ")

print(hamming\_distance(s1,s2))