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

# Python 3 program to find the sum of numbers

# divisible by M in the given range

# Function to find the sum of numbers

# divisible by M in the given range

def sumDivisibles(A, B, M):

# Variable to store the sum

sum = 0

# Running a loop from A to B and check

# if a number is divisible by i.

for i in range(A, B + 1):

# If the number is divisible,

# then add it to sum

if (i % M == 0):

sum += i

# Return the sum

return sum

# Driver code

if \_\_name\_\_=="\_\_main\_\_":

# A and B define the range

# M is the dividend

A = 6

B = 15

M = 3

# Printing the result

print(sumDivisibles(A, B, M))

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 check(s):

regex=eval(s)

if regex:

return True

else:

return False

# printing result

print(check("3 < 7 < 11"))

print(check("13 > 44 > 33 > 1"))

print(check("1< 2< 6< 9 > 3"))

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

# Function to Replace each vowel in

# the string with a specified character

def replaceVowelsWithK(test\_str, K):

# string of vowels

vowels = 'AEIOUaeiou'

# iterating to check vowels in string

for ele in vowels:

# replacing vowel with the specified character

test\_str = test\_str.replace(ele, K)

return test\_str

# Driver Code

# input string

input\_str = "Geeks for Geeks"

# specified character

K = "#"

# printing input

print("Given String:", input\_str)

print("Given Specified Character:", K)

# printing output

print("After replacing vowels with the specified character:",

replaceVowelsWithK(input\_str, K))

**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 hammingDist(str1, str2):

i = 0

count = 0

while(i < len(str1)):

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

count += 1

i += 1

return count

# Driver code

str1 = "geekspractice"

str2 = "nerdspractise"

# function call

print(hammingDist(str1, str2))