**BASIC PROGRAMMING ASSIGNMENT\_17-SUBMITTED BY SAMUEL DEVDAS**

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

Ans.

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

total=[]

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

if num%c==0:

total.append(num)

return(sum(total))

evenly\_divisible(1,10,3)

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

Ans.

def correct\_signs(exp):

return exp

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

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\*"

Ans.

def replace\_vowels(a,b):

vowels=['a','e','i','o','u']

for elem in vowels:

a=a.replace(elem,b)

print(a)

replace\_vowels("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

**Ans.**

**def factorial(n):**

**if n==1:**

**return n**

**elif n== 0:**

**print("The factorial of 0 is 1")**

**else:**

**return n\*factorial(n-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

Ans.

def hamming\_distance(str1,str2):

alphabets=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

difference=[]

for i in range(len(str1)):

difference.append(abs(alphabets.index(str1[i])-alphabets.index(str2[i])))

return('Hamming distance is:',sum(difference))

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