**ADITYA AMIN**

**ASSIGN : 17**

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 evenly\_divisible(a, b, c):

sum = 0

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

if num % c == 0:

sum += num

return sum

print(evenly\_divisible(1, 10, 20))

# Output: 0

print(evenly\_divisible(1, 10, 2))

# Output: 30

print(evenly\_divisible(1, 10, 3))

# Output: 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

def correct\_signs(expr):

# Split the expression into a list of tokens

tokens = expr.split()

# Iterate over the tokens pairwise, checking the inequality between each pair

for i in range(0, len(tokens) - 2, 2):

if tokens[i+1] == "<" and int(tokens[i]) >= int(tokens[i+2]):

return False

elif tokens[i+1] == ">" and int(tokens[i]) <= int(tokens[i+2]):

return False

# If we haven't found any incorrect inequalities, the expression is correct

return True

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

# Output: True

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

# Output: False

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

# Output: 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\*"

def replace\_vowels(txt, ch):

vowels = 'aeiouAEIOU'

for letter in txt:

if letter in vowels:

txt = txt.replace(letter, ch)

return txt

#Example usage:

print(replace\_vowels("the aardvark", "#")) # Output: "th# ##rdv#rk"

print(replace\_vowels("minnie mouse", "?")) # Output: "m?nn?? m??s?"

print(replace\_vowels("shakespeare", "")) # Output: "shksp\*\*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

def factorial(n):

if n == 0:

return 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

def hamming\_distance(txt1, txt2):

if len(txt1) != len(txt2):

return "Strings have different lengths, cannot compute Hamming distance."

else:

return sum([1 for i in range(len(txt1)) if txt1[i] != txt2[i]])

example cases

print(hamming\_distance("abcde", "bcdef")) # expected output: 5

print(hamming\_distance("abcde", "abcde")) # expected output: 0

print(hamming\_distance("strong", "strung")) # expected output: 1