**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 add(a,b,c):  
 j=0  
 for i in range(a,b+1):  
 if(i%c==0):j=j+i  
 return j  
a=int(input())  
b=int(input())  
c=int(input())  
print(add(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 check(s):  
 regex=eval(s)  
 if regex:  
 return True  
 else:  
 return False  
s=input()  
print(check(s))

**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\*"**

string=input("Enter the string: ")  
char=input("Enter the char")  
n\_str=''  
for i in range(len(string)):  
 if(string[i] in ['a','e','i','o','u','A','E','I','O','U']):  
 n\_str=n\_str+char  
 else:n\_str=n\_str+string[i]  
print(n\_str)

**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 fact(n):  
 if(n<=0):print("Enter a valid no")  
 elif(n==1):  
 return n  
 else:  
 return n\*fact(n-1)  
n=int(input())  
print(fact(n))

**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 ham\_dist(str1,str2):

i=0

count=0

while(i<len(str1)):

if(str1[i]!=str2[i]): count+=1

i+=1

return count

str1=input()

str2=input()

print(ham\_dist(str1,str2))