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

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

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

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

Answer 1:

Here is the 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.

arduino
```
def evenly_divisible(a, b, c):
    total = 0
    for num in range(a, b+1):
        if num % c == 0:
            total += num
    return total
```

Examples:

bash
```
print(evenly_divisible(1, 10, 20)) # 0
print(evenly_divisible(1, 10, 2)) # 30
print(evenly_divisible(1, 10, 3)) # 18
```

Output:

```
0
30
18
```

Answer 2:

Here is the function that takes a given inequality expression and returns True if it is correct and False otherwise.

python
```
def correct_signs(expression):
    return eval(expression)
```

Examples:

bash
```
print(correct_signs("3 < 7 < 11")) # True
print(correct_signs("13 > 44 > 33 > 1")) # False
print(correct_signs("1 < 2 < 6 < 9 > 3")) # True
```

Output:

graphql
```
True
False
```

```
True
```

Answer 3:

Here is the function that replaces all the vowels in a string with a specified character.

c
```
def replace_vowels(string, character):
    vowels = "aeiouAEIOU"
    new_string = ""
    for letter in string:
        if letter in vowels:
            new_string += character
        else:
            new_string += letter
    return new_string
```

Examples:

bash
```
print(replace_vowels("the aardvark", "#")) # "th# ##rdv#rk"
print(replace_vowels("minnie mouse", "?")) # "m?nn?? m??s?"
print(replace_vowels("shakespeare", "*")) # "sh*k*sp**r*"
```

Output:

less
```
th# ##rdv#rk
m?nn?? m??s?
sh*k*sp**r*
```

Answer 4:

Here is the function that calculates the factorial of a number recursively.

arduino
```
def factorial(num):
    if num == 0:
        return 1
    else:
        return num * factorial(num-1)
```

Examples:

bash
```
print(factorial(5)) # 120
print(factorial(3)) # 6
print(factorial(1)) # 1
print(factorial(0)) # 1
```

Output:

```
120
6
1
1
```

Answer 5:

Here is the function that computes the Hamming distance between two strings.

python
```
def hamming_distance(str1, str2):
    if len(str1) != len(str2):
        return None
    else:
        count = 0
        for i in range(len(str1)):
            if str1[i] != str2[i]:
                count += 1
        return count
```

Examples:

bash
```
print(hamming_distance("abcde", "bcdef")) # 5
print(hamming_distance("abcde", "abcde")) # 0
print(hamming_distance("strong", "strung")) # 1
```

Output:

```
5
0
1
```