Programming_Assingment17

```
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
```

In [2]:

```
def sumDivisibles(a, b, c):
    sum = 0
    for i in range(a, b + 1):
        if (i % c == 0):
            sum += i
    return sum
a = int(input('Enter a : '))
b = int(input('Enter b : '))
c = int(input('Enter c : '))
print(sumDivisibles(a, b, c))
```

```
Enter a : 1
Enter b : 10
Enter c : 3
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

In [3]:

```
def correct_signs ( txt ) :
    return eval ( txt )
print(correct_signs("3 > 7 < 11"))
print(correct_signs("13 > 44 > 33 > 1"))
print(correct_signs("1 < 2 < 6 < 9 > 3"))
```

```
False
False
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*'

In [4]:

```
def replace_vowels(str, s):
    vowels = 'AEIOUaeiou'
    for ele in vowels:
        str = str.replace(ele, s)
    return str

input_str = input("enter a string : ")
s = input("enter a vowel replacing string : ")
print("\nGiven Sting:", input_str)
print("Given Specified Character:", s)
print("Afer replacing vowels with the specified
character:",replace_vowels(input_str, s))
```

```
enter a string : akash
enter a vowel replacing string : @

Given Sting: akash
Given Specified Character: @
Afer replacing vowels with the specified character: @k@sh
```

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

In [5]:

```
def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n-1)

num = int(input('enter a number :'))
print("Factorial of", num, "is", factorial(num))
```

```
enter a number :5
Factorial of 5 is 120
```

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

In [6]:

```
def hamming_distance(str1, str2):
    i = 0
    count = 0

    while(i < len(str1)):
        if(str1[i] != str2[i]):
            count += 1
        i += 1
    return count

# Driver code
str1 = "abcde"
str2 = "bcdef"

# function call
print(hamming_distance(str1, str2))
```

```
5
```

In [7]:

```
print(hamming_distance('strong', 'strung'))
```

```
1
```

In [8]:

```
hamming_distance('abcde', 'abcde')
```

Out[8]:

```
0
```