

UNIVERSIDAD ESCUELA COLOMBIANA DE INGENIERÍA

Asignatura: Algoritmos y Programación, grupo 16 (AYPR-16)

Profesora: Ingeniera Patricia Salazar Perdomo

Fecha: 20 de febrero de 2024

Ref.: Tarea No. 10. Programación modular en Python.

EN LA PRÓXIMA CLASE HABRÁ PRUEBA CORTA SOBRE PROGRAMACIÓN MODULAR.

Construya las siguientes **funciones en Python**, una para cada problema, y **hágales pruebas suficientes**. Los argumentos y los parámetros deben corresponder a lo que se necesita para hacer cada tarea. Escriba un ejemplo de invocación para cada una. Observe los ejemplos que se dan al final. También le pueden servir los que están en la carpeta EJEMPLOS - UNA función con argumentos y parámetros.

1. Averigua si tres longitudes, que ya se conocen, pueden corresponder a los lados de un triángulo. Corresponderán si la suma de dos lados cualesquiera es mayor que la medida del tercer lado. La función retornará 1 si se trata de un triángulo o 0 de lo contrario.
2. Averigua la suma de los divisores impares positivos de un número entero positivo que ya se conoce.

Ejemplo 1. Número: 60

Resultado. La suma de los divisores impares de 60 es 24.

Nota: Los divisores impares positivos de 60 son 1, 3, 5 y 15. No los tiene que escribir.

Ejemplo 2. Número: 21

Resultado. La suma de los divisores impares de 21 es 32.

Nota: Los divisores impares positivos de 21 son 1, 3, 7 y 21. No los tiene que escribir.

EJEMPLOS

Ejemplo 1. Función que obtiene la suma de los números desde n hasta 1, de 2 en 2 y en forma descendente. El valor de la suma no lo debe escribir en la función, pero debe quedar disponible en la función que invoque o llame a ésta.

def suma_desc_2_en_2 (n):

```
    suma = 0
    valor = n
    while valor > 0:
        suma = suma + valor
        valor = valor - 2
    return suma
```

#Ejemplo de invocación 1.

```
n = 41
suma = suma_desc_2_en_2 (n)
print ("Suma =", suma) # Para verificar el resultado.
```

#Ejemplo de invocación 2.

```
suma = suma_desc_2_en_2 (20) # La suma la hará desde 20 hasta 2, inclusive.
print ("Suma =", suma) # Para verificar el resultado.
```

Ejemplo 2. Función que escribe las primeras n potencias de 2.

def primeras_potencias_de_2 (n):

```
    cont_pot2 = 1
    pot2 = 1
    print ("\nPrimeras", n, "potencias de 2:\n")
    while cont_pot2 <= n:
        print (pot2, end = " ")
        pot2 = pot2 * 2
        cont_pot2 = cont_pot2 + 1
```

#Ejemplo de invocación.

```
primeras_potencias_de_2 (10) # Imprimirá las primeras 10 potencias de 2.
```

Anexo los archivos de los ejemplos.