



## 503203/503201/503215 Programación Programación Usando Funciones

Javier Vidal

14 de mayo de 2022

1.- Los *números amigos* son dos enteros positivos tales que la suma de los divisores propios<sup>1</sup> de uno de ellos es igual al otro. Por ejemplo, el par 220 y 284 son números amigos ya que:

- los divisores propios de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, que suman 284
- y los divisores propios de 284 son 1, 2, 4, 71 y 142, que suman 220

Se pide, escriba un programa en Python que reciba un número entero positivo  $n$  y que calcule y despliegue un número *amigo* si es que existe.

**Entrada:** La única entrada a este programa es un número entero  $n$  ( $n > 0$ ).

**Salida:** La única salida de este programa es un número entero  $m$  ( $m > 0$ ), tal que  $m$  es *amigo* de  $n$ . Si no existe  $m$  debe desplegar el mensaje “ $n$  no tiene amigo”, donde  $n$  representa el valor en la entrada.

**Ejemplo de Entrada:** 220

**Ejemplo de Salida:** 284

**Observación:** Utilice funciones para desarrollar este ejercicio

SOLUCIÓN:

```
def validar(inf, sup, msg):
    val=int(input(msg))
    if inf < sup:
        while val < inf or val > sup:
            val=int(input(msg))
    else:
        while val < inf:
            val=int(input(msg))
    return val

def suma_divisores_propios(x):
    suma = 1
    i = 2
    while i <= x // 2:
        if x % i == 0:
            suma = suma + i
        i = i + 1
    return suma
```

---

<sup>1</sup>El 1 se considera divisor propio, pero no lo es el mismo número

---

```

# programa principal
n = validar(0,-1,'n=')
suma_dp_n = suma_divisores_propios(n)
m = suma_dp_n
suma_dp_m = suma_divisores_propios(m)
if n == suma_dp_m and m == suma_dp_n:
    print(n,' y ', m, ' son amigos')
else:
    print(n, ' no tiene amigos')
# fin

```

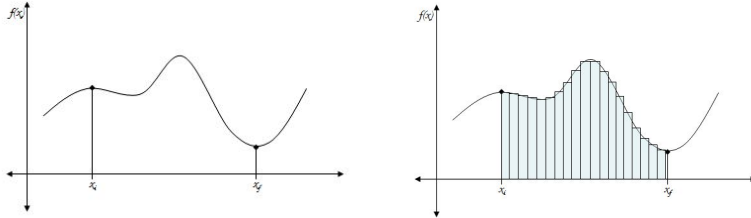
2.- El resultado del cálculo de una integral

$$\int_{x_i}^{x_f} f(x)dx \quad (1)$$

se puede aproximar calculando la sumatoria:

$$\sum_{j=0}^n \Delta f(x_j)dx \quad (2)$$

con  $\Delta = \frac{x_f - x_i}{n}$  y  $x_j = x_i + \Delta \cdot j$ . Gráficamente esto se puede esquematizar de la siguiente manera:



Se pide: construya un programa en Python que lea los valores de  $x_i$ ,  $x_f$  y  $n$  y que calcule la integral de una función como  $\text{sen}(x) + \text{cos}(x)$ .

SOLUCIÓN:

```

def validar(inf, sup, msg):
    val=int(input(msg))
    if inf < sup:
        while val < inf or val > sup:
            val=int(input(msg))
    else:
        while val < inf:
            val=int(input(msg))
    return val

# Función que calcula el factorial de un número entero n n>=0
def factorial(n):
    res=1

```

---

```

    while n>0:
        res = res * n
        n = n - 1
    return res

# Función que calcula el seno de un ángulo x expresado en radianes
def seno(x):
    suma = 0
    for i in range(0, n+1):
        suma = suma + (-1)**i*x**(2*i+1)/factorial(2*i+1)
    return suma

# Función que calcula el coseno de un ángulo x expresado en radianes
def coseno(x):
    suma = 0
    for i in range(0, n+1):
        suma = suma + (-1)**i*x**(2*i)/factorial(2*i)
    return suma

def f(x):
    return seno(x)+coseno(x)

def integral (xi, xf, n):
    j = 0
    delta = (xf-xi)/n
    suma = 0
    while j <= n:
        xj = xi + delta * j
        suma = suma + delta * f(xj)
        j = j + 1
    return suma

# programa principal
xi = float(input('x_inicial='))
xf = float(input('x_final='))
n = validar(1,0,'n=')
print(integral(xi, xf, n))
# fin

```