



Semana-07

Programación modular



Temario



- Uso de dos o mas funciones para resolver un problema



PROGRAMACIÓN MODULAR



Usar una o mas
funciones para
implementar un
programa

Paso 1: Implementación de funciones



Ejemplo:

Implementar una función que permita retornar el resultado de convertir a radianes un ángulo en grados enviado como argumento.

¿Qué se espera de la función cuando sea invocada?

0.5237 ← convierteRadianes(30)

0.7854 ← convierteRadianes(45)

1.0472 ← convierteRadianes(60)

Ejemplo:



Implementar una función que permita retornar el factorial de un número enviado como argumento

¿Qué se espera de la función cuando sea invocada?

1	←	<code>factorial(0)</code>
24	←	<code>factorial(4)</code>
3628800	←	<code>factorial(10)</code>
9.3326e+157	←	<code>factorial(100)</code>

Ejemplo:



Implementar una función que permita retornar el seno de un ángulo en radianes, el cual será enviado como argumento.

Para la implementación tendrá que usar la serie de Tylor:

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

El número de términos a utilizar en la sumatoria de la serie también será enviado como argumento a la función

¿Qué se espera de la función cuando sea invocada?

0.5 ← `miFuncionSeno(pi/6, 10)`

0.707 ← `miFuncionSeno(pi/4, 10)`



factorial

miFuncionSeno

El código de la función ***miFuncionSeno*** tendrá que invocar un número determinado de veces a la función ***factorial***

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$



Paso 2: PROBLEMA:

Implementar un programa que le permita al usuario ingresar un ángulo en grados, y luego de ello se muestre en pantalla el resultado de calcular el seno del ángulo usando la serie de Taylor para una cantidad de términos de 1, 2, 3, 4, 20.

Ejemplo:

```
Ingresa angulo en grados: 30
sen(30) con 1 terminos: 0.5236
sen(30) con 2 terminos: 0.499675
sen(30) con 3 terminos: 0.500003
sen(30) con 4 terminos: 0.500001
sen(30) con 5 terminos: 0.500001
sen(30) con 6 terminos: 0.500001
sen(30) con 7 terminos: 0.500001
sen(30) con 8 terminos: 0.500001
sen(30) con 9 terminos: 0.500001
sen(30) con 10 terminos: 0.500001
sen(30) con 11 terminos: 0.500001
sen(30) con 12 terminos: 0.500001
sen(30) con 13 terminos: 0.500001
sen(30) con 14 terminos: 0.500001
sen(30) con 15 terminos: 0.500001
sen(30) con 16 terminos: 0.500001
sen(30) con 17 terminos: 0.500001
sen(30) con 18 terminos: 0.500001
sen(30) con 19 terminos: 0.500001
sen(30) con 20 terminos: 0.500001
```




convierteRadianes

factorial

miFuncionSeno

Función main:

1. Leer el ángulo en grados:
2. Convertir el ángulo en radianes
3. Calcular el seno del ángulo para 20 cantidades de términos de la serie de Taylor.

Invocar 20 veces a la función *miFuncionSeno*

DIAGRAMA GENERAL DE LA APLICACION:

