

Casos de Estudio

Analice, diseñe y codifique los siguientes enunciados en Python

1. Diseñar un algoritmo que ordene tres números a, b, c en forma ascendente utilizando un módulo denominado menorMayor que tiene dos parámetros y que devuelve en el primer parámetro el valor menor y en el segundo el valor mayor de los parámetros respectivamente.
2. Analizar y ejecutar los ejemplos de los programas en Python que utilizan las variables libres (variables globales y no locales) para cada uno de los casos propuestos.

Ejercicios

1. Analizar y ejecutar el algoritmo que calcula la serie de Taylor de la función seno de más abajo, donde x es el valor de un ángulo (expresado en radianes) y n es el número de términos. Mostrar el esquema de los módulos según la programación modular, y los ámbitos de las variables. También hacer la prueba de escritorio para un ángulo de 45 grados sexagesimales y 4 términos, utilice el *debugger* del VSC.

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}, \forall x$$

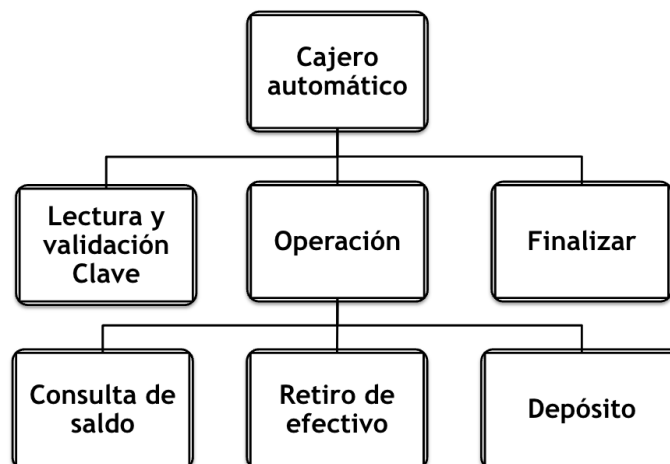
<pre>def bienvenida (n): nro = 20 car = '.' print(car*nro) print('Hola ', n) print(car*nro) print('Este algoritmo calcula el seno ') print('de un ángulo en grado sexagesimal') print('con la serie de Taylor hasta un ') print('término determinado por Ud. ') print(car*nro) def verificaP(): while True: x = float(input('Ángulo en sexagesimales x>=0:')) if x >= 0: return x def verificaMenor10(): while True: m10 = int(input('N° de términos [1,10]:')) if m10 >= 1 and m10 <= 10: return m10</pre>	<pre>def factorial(x): p = 1 for i in range(1, x+1): p *= i return p def potencia(base, exponente): p = 1 for i in range(exponente): p *= base return p #Principal nombre = input('Ingrese su nombre:') bienvenida(nombre) a = verificaP() x = a*pi/180 #radianes m = verificaMenor10() #valor final términos suma = 0 for n in range(m): t = 2*n+1 suma += potencia(-1,n)/factorial(t)*potencia(x,t) print('Seno calculado = ', suma) print('Seno función interna = ',sin(x))</pre>
---	--

2. Escribir un módulo denominado PMS que tiene dos parámetros formales base y exponente. Calcular la base elevada al exponente, siendo la base un número real cualquiera y exponente un valor entero positivo o nulo. Utilizar las multiplicaciones sucesivas de la base. Si el cálculo no puede realizarse debe devolver cero (0).
3. Mediante un menú de opciones resolver:
 - a. Ingresar dos valores a y b validando que estén en el intervalo [0,9] y mostrarlos en letras separados por una serie de asteriscos
 - b. Si a y b son pares intercambiar los valores y mostrarlos en letras
 - c. Salir

4. Analizar, ejecutar, realizar la prueba de escritorio en forma manual y con el debugger del VSC.

Pruebe para s = 1	Pruebe para v = 10, luego elimine el # de la última línea, debe mostrar el tiempo. ¿Cómo evita utilizar global v?
<pre>def dos(s): s = s + 5 print('Dos:', s) def uno(s): s = s + 10 dos(s) print('Uno:', s) return s #principal s = int(input('Inicial:'))#1 s = s + uno(s) print('Global:',s)</pre>	<pre>def acelerar(): global v tiempo = 1 v+= 5 return # programa principal v = float(input('Velocidad?:')) print(f'Velocidad: {v} km/h') print('Aumento la velocidad!') acelerar() print(f'Velocidad: {v} km/h') #print('Tiempo:', tiempo)</pre>

5. Diseñar un algoritmo que simule una calculadora básica debe utilizar un módulo denominado calcular(op,a,b) dónde los parámetros a y b son variables de tipo float y representan los operandos de la expresión aritmética y op es un parámetro de tipo cadena de caracteres que puede ser: 'suma', 'resta', 'multiplicación' y 'división', y se realiza la operación correspondiente. Mediante un menú de opciones el operador debe ingresar los datos de a y b y luego debe poder seleccionar una operación aritmética.
6. Escribir un algoritmo que simule las operaciones que se realizan en un cajero automático de acuerdo al esquema jerárquico de los módulos que se nota en la figura:



7. Un tanque de agua tiene una capacidad fija en litros que no puede cambiar. Se puede cargar y vaciar un volumen determinado de agua en litros. No se puede llenar más de su capacidad así como tampoco se puede vaciar más del volumen que posee. Se pide implementar un algoritmo que mediante un menú de opciones y de manera modular que permita cargar o descargar una cantidad permitida de agua solicitado por el operador, también se debe poder consultar el volumen de agua en el tanque.
8. Existe una plataforma donde se colocan tanques de agua. Para esta plataforma se quiere implementar un algoritmo denominado: controlador_de_tanques que realiza las mismas operaciones que el algoritmo del punto anterior. Al crearse el controlador_de_tanques se debe determinar la cantidad máxima de tanques que puede controlar de 2 a 4. El controlador_de_tanques permite agregar y quitar tanques de agua. Se desea saber cuántos tanques de agua hay, el volumen ocupado con agua y la capacidad vacía que tiene cada tanque.