

Casos de Estudio de Modularización

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, realizar la prueba de escritorio y mostrar la salida de los siguientes programas:

i = 10	a = 7, b = 9, c = 3	a = 3, b = 9, c = 7
<pre>i = int(input('i:')) if (i == 10): if (i < 15): print('uno') if (i < 12): print('dos') else: print('tres')</pre>	<pre>a = int(input('a:')) b = int(input('b:')) c = int(input('c:')) if (a>b and a>c) and (a != b and a != c): print(a) elif (b>a and b>c) and (b != a and b != c): print(b) elif (c>a and c>b) and (c != a and c != b): print(c) else: print("iguales")</pre>	<pre>a = int(input('a:')) b = int(input('b:')) c = int(input('c:')) x = a != b y = not c >= a and not b == 0 z = not x or y if not x or y: print('alternativa 1') elif not x or not y and z: print('alternativa 2') elif not x and not z or y: print('alternativa 3') else: print('alternativa 4')</pre>

2. Analizar, realizar la prueba de escritorio y ejecutar en python cada uno de los casos del algoritmo de la serie par, $s = 2 + 4 + 6 + 8 + 10 + 12 + 14 + \dots$. Mostrar en cada caso el valor de la suma s y el valor del término i.
 - a. hasta que la acumulación de términos sea mayor que x, para x=25.
 - b. hasta el n-ésimo término, para n=4.
 - c. hasta que el n-ésimo término sea mayor o igual que q, para q=13.

<pre>#serie_par_a x = int(input('valor a superar:')) s = 0 i = 0 while s <= x: i += 1 s += 2 * i print(f'serie={s} términos={i-1}')</pre>	<pre>#serie_par_b n = int(input('nro términos:')) s = 0 for i in range(1, n+1): s += 2 * i print(f'serie={s} términos={i}')</pre>	<pre>#serie_par_c q = int(input('término a superar:')) s = 0 i = 1 t = 2 * i while t < q: s += t i += 1 t = 2 * i print(f'serie={s} términos={i}')</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Se leen repetidamente valores de v (carácter) y w (entero). Si v es igual a 'A' entonces mostrar si w es positivo o no lo es, si v es igual a 'B' calcular y mostrar siempre que sea posible $f = \sqrt{\frac{1}{w}}$ si v es igual a 'C' mostrar el mensaje 'Ok'. El ingreso de datos finaliza cuando w es igual a cero. Se desea además mostrar cuántos valores de v no fueron ni 'A' ni 'B' ni 'C'.

4. Resolver con papel y lápiz y luego controlar en python.

<p>Reescribir el siguiente algoritmo reemplazando la estructura repetitiva for range, con la estructura while, de manera que el algoritmo resuelva el mismo problema.</p> <pre> n = int(input('n: ')) s = 0 for i in range(3, n, 2): s += i print(s) </pre>	<p>Hacer la prueba de escritorio del siguiente algoritmo, para n = 6 y mostrar el valor final de s.</p> <pre> n = int(input('n: ')) i = 3 s = 0 f1 = 1 f2 = 1 while i <= n: f3 = f1 + f2 f1 = f2 f2 = f3 if i % 2 != 0: s += f3 i += 1 print(s) </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5. Los tres lados a, b y c de un triángulo deben satisfacer la [desigualdad triangular](#): cada uno de los lados no puede ser más largo que la suma de los otros dos. Escriba un programa que reciba como entrada los tres lados de un triángulo, e indique: si acaso el triángulo es inválido; y si no lo es, qué tipo de triángulo es (equilátero, isósceles, escaleno)
6. Escriba un programa que mediante un menú de opciones: ingrese la fecha de nacimiento validando los datos, muestre la edad del usuario y una opción de despedida con el cierre del programa. El programa debe tener en cuenta si el cumpleaños ingresado ya pasó durante este año, o si todavía no ocurre.

<p>Ingrese su fecha de nacimiento. Por supuesto, el resultado entregado depende del día en que su programa será ejecutado (21/04/2023)</p>	<p>Para obtener la fecha actual, puede hacerlo usando la función localtime que viene en el módulo time.</p>
<p>dd: 14 mm: 6 aaaa: 1948 Su edad es: 75 años Falta para su cumpleaños 2 meses</p>	<pre> from time import localtime t = localtime() print(t.tm_mday) #día print(t.tm_mon) #mes print(t.tm_year) #año </pre>

7. Realizar la multiplicación de dos números enteros n1 y n2 mediante sumas sucesivas, hacer dos algoritmos uno con la estructura while y otro con la estructura for. Finalmente haga un algoritmo que multiplique tres factores mediante sumas sucesivas. ¿Cómo haría un algoritmo que multiplique muchos factores con el mismo método anterior?
8. Escribir un programa que consulte a una cantidad n de estudiantes, el nombre y las notas de las asignaturas de un curso (Análisis Matemático, Álgebra, Introducción a la Programación, e Inglés). Pregunte al usuario la nota que ha obtenido en cada asignatura. Mostrar para cada estudiante, cuántas asignaturas aprobó (nota mayor a 50 incluido), y si la nota es menor a 30 debe alertar al estudiante que “ES MUY PROBABLE QUE DEBA RECURSAR”. Al final el programa debe mostrar el nombre del estudiante con el promedio más alto; el nombre del estudiante y la materia de aquel que tiene la nota mínima de todo el curso.
9. Se quiere determinar el importe a facturar a los clientes de unos grandes almacenes según estos criterios:



-
- a. Si pagan con tarjeta oro tendrán un 15% de descuento.
 - b. Si pagan con tarjeta club tendrán un 5% de descuento.
 - c. Si la tarjeta (oro o club) es modalidad joven, tendrán un 5% plus de descuento.
10. Se quiere determinar la nómina de los empleados de una empresa de acuerdo con estos criterios:
- a. Si el empleado es altamente productivo tendrá en nómina un plus de productividad.
 - b. Si el empleado es encargado de su grupo tendrá en nómina un plus de encargado.
 - c. Si el empleado ha cometido una infracción grave durante ese mes le será eliminado cualquier plus que pudiera tener.