

Casos de Estudio

Ejercicios

- 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>

- 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 .
 - hasta que la acumulación de términos sea mayor que x , para $x=25$.
 - hasta el n -ésimo término, para $n=4$.
 - 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>
--	---	---

- 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)
- 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.

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

5. ¿Puede explicar con sus propias palabras por qué ambos programas imprimen lo mismo?

Argumentos posicionales	Argumentos de palabras clave
<pre>def nombre_persona(nombre, apellido): print(f'{nombre} {apellido}') nombre_persona('Lionel', 'Messi')</pre>	<pre>def nombre_persona(nombre, apellido): print(f'{nombre} {apellido}') nombre_persona(apellido = 'Messi', nombre = 'Lionel')</pre>

6. Analice y realice la prueba de escritorio utilizando el *debugger* de VSC

<pre>def mi_funcion(x, y=50): """Parámetros con valores iniciales""" print('x:', x) print('y:', y) a = int(input('a:')) #10 mi_funcion(a) print(mi_funcion.__doc__)</pre>	<pre>def f1(): """Juan Galan: poeta jujeño 1913-1963""" s = "Jujuy le han puesto de nombre, debe ser cosa de Dios; en el idioma del cielo así se llama el amor..." def f2(): print(s) f2() f1() print(f1.__doc__)</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 módulo que utiliza tres parámetros que representan un ancho y altura, el tercer parámetro es un carácter a utilizar en el dibujo de un rectángulo. En el ejemplo siguiente se leen los valores 5 (ancho), 3 (altura) con el carácter "o", resultando el gráfico:
- ```
o o o o o
o o o o o
o o o o o
```
9. 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.

10. Realice dos algoritmos para calcular la siguiente fórmula:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

- Realice el algoritmo para una cantidad n de términos de la suma.
- Realice el algoritmo hasta que el término o error o aproximación sea menor a 0,001

11. Realizar un programa modular que muestre un menú de opciones. Debe ingresar inicialmente dos valores que representan el numerador y denominador de una fracción.

- sumar\_fracciones(n1,d1,n2,d2) calcula la suma de dos fracciones. El resultado es otra fracción cuyo numerador es  $n1*d2+d1*n2$  y denominador  $d1*d2$ . Se debe simplificar la fracción resultado.
- restar\_fracciones(n1,d1,n2,d2) calcula la resta de dos fracciones. El resultado es otra fracción  $\text{numerador}=n1*d2-d1*n2$  y  $\text{denominador}=d1*d2$ . Se debe simplificar la fracción resultado.
- multiplicar\_fracciones(n1,d1,n2,d2) calcula el producto de dos fracciones. El resultado es una fracción con numerador  $n1*n2$  y denominador  $d1*d2$ . Se debe simplificar la fracción resultado.
- dividir\_fracciones(n1,d1,n2,d2) calcula la división de dos fracciones. El resultado es una fracción con numerador  $n1*d2$  y denominador  $d1*n2$ . Se debe simplificar la fracción resultado.

En cada ejecución del programa deberá ingresar la fracción solicitando numerador y denominador. La fracción debe simplificarse y mostrarse.

Al mostrar la fracción se debe mostrar la misma simplificada. En caso de denominador 1, sólo debe mostrar el numerador.

Puede implementar una función `simplificar_funcion(n,d)` que devuelve 2 nuevos valores resultados de la simplificación, para ello hay que dividir n y d por el mcd de ambos.

Puede implementar una función `mcd(n,d)` que devuelve el máximo común divisor de ambos parámetros.

12. Se quiere determinar el importe a facturar a los clientes de unos grandes almacenes según estos criterios:

- Si pagan con tarjeta oro tendrán un 15% de descuento.
- Si pagan con tarjeta club tendrán un 5% de descuento.
- Si la tarjeta (oro o club) es modalidad joven, tendrán un 5% plus de descuento.

13. Se quiere determinar la nómina de los empleados de una empresa de acuerdo con estos criterios:

- Si el empleado es altamente productivo tendrá en nómina un plus de productividad.
- Si el empleado es encargado de su grupo tendrá en nómina un plus de encargado.
- Si el empleado ha cometido una infracción grave durante ese mes le será eliminado cualquier plus que pudiera tener.