

TALLER DE FUNCIONES

Crear un archivo calculos.py, con las siguientes funciones

Función determinarResultadosIMC

Parámetros			
	imc	Float	El índice de masa corporal
Funcionalidad			
	<p>Retorna un mensaje, indicando los resultados del análisis del imc de la persona, de acuerdo a la siguiente tabla:</p> <p>Delgadez severa: [0 - 16[Delgadez moderada: [16-17[Delgadez leve: [17-18.5[Peso normal: [18.5-25[Sobrepeso: [25-30[Obesidad Grado 1: [30-35[Obesidad Grado 2: [35-40[Obesidad Grado 3: Mayor o igual a 40</p> <p>IMPORTANTE: Considerar los valores INCLUIDOS y EXCLUIDOS en cada rango</p> <p>Ejemplo de invocación: resultado=determinarMasa(16) resultado tiene el mensaje Delgadez moderada</p>		
Retorno			
	El mensaje con el resultado del análisis. Si el valor del imc no está en ningún rango, responder IMC fuera de rango		
Test			
	Archivo: testIMC.py		
	Resultados esperados:		

	<pre> imc: 0 => Delgadez severa imc: 16 => Delgadez moderada imc: 16.9 => Delgadez moderada imc: 17 => Delgadez leve imc: 35 => Obesidad Grado 2 imc: 40 => Obesidad Grado 3 imc: 27.5 => Sobrepeso imc: 45 => Obesidad Grado 3 imc: -2 => IMC fuera de rango imc: 0 => Delgadez severa imc: 16 => Delgadez moderada imc: 16.9 => Delgadez moderada imc: 17 => Delgadez leve imc: 35 => Obesidad Grado 2 imc: 40 => Obesidad Grado 3 imc: 27.5 => Sobrepeso imc: 45 => Obesidad Grado 3 imc: -2 => IMC fuera de rango </pre>	
--	--	--

Función encontrarMayor

Parámetros			
	valor1	int	Valor 1
	valor2	int	Valor 2
	valor3	int	Valor 3
Funcionalidad			
	<p>Para determinar el mayor, puede utilizar el siguiente algoritmo: Crear una variable llamada mayorActual, inicialmente se le asigna valor1, es decir: mayorActual = valor1 En esta variable siempre estará guardado el mayor de los números, como aún no analizamos cuál es mayor, le asignamos directamente valor1.</p> <p>Luego empezamos a comparar los valores restantes, por ejemplo valor2 contra mayorActual, y si valor2 es mayor, se le asigna a la variable, así: mayorActual=valor2.</p> <p>Luego comparar con valor3 y aplicar la misma lógica.</p> <p>De esta forma aseguramos que siempre en la variable mayorActual, se esté asignando el valor mayor que puede ser valor1, valor2 o valor3.</p> <p>Al final de todas las comparaciones retornar mayorActual</p>		

Retorno	
	El valor mayor.
Test	
	Archivo: testMayor.py
	Resultados esperados: <pre> El mayor entre 10, 20 y 30 es 30 El mayor entre 100, 200 y 50 es 200 El mayor entre 100, 50 y 90 es 100 El mayor entre 40, 50 y 50 es 50 El mayor entre 80, 80 y 20 es 80 </pre>

Función encontrarMenor

Parámetros			
	valor1	int	Valor 1
	valor2	int	Valor 2
	valor3	int	Valor 3
	valor4	int	Valor 4
Funcionalidad			
	Para determinar el menor, aplicar un algoritmo similar al de encontrarMayor, pero esta vez crear una variable menorActual, donde se irá guardando el menor valor.		
Retorno			
	El valor menor.		
Test			
	Archivo: testMenor.py		
	Resultados esperados: <pre>El menor entre 10, 20, 30 y 40 es 10 El menor entre 100, 200, 50 y 0 es 0 El menor entre 100, 50, 30 y 40 es 30 El menor entre 20, 10, 5 y 5 es 5 El menor entre 100, 80, 90 y 200 es 80</pre>		

Función calcularEdad

Parámetros		
	anioNacimiento	int Es el año en el cual nació la persona
Funcionalidad		
	<p>Esta función va a calcular la edad que cumplirá el usuario, en el año en curso. Para esto, seguir los siguientes pasos: Obtener el año actual y guardar en una variable Validar que el anioNacimiento, sea mayor o igual que cero y menor o igual al año actual. Si está dentro del rango, calcular la edad, caso contrario retornar -1. Si el anioNacimiento, está dentro del rango, calcular la edad. Para calcular la edad, restar el año actual menos el año de nacimiento y retornar dicho valor.</p>	
Retorno		
	La cantidad de años que el usuario cumplirá en el año actual. Si el año está fuera de rango, retorna -1	
Test		
	Archivo: testEdad.py	
	<p>Resultados esperados:</p> <pre>Año nacimiento: 2050 => Usted aun no nace, o nació antes de cristo Año nacimiento: -10 =>Usted aun no nace, o nació antes de cristo Año nacimiento: 2024 =>Este año cumple 0 años Año nacimiento: 2013 =>Este año cumple 11 años</pre>	

Función calcularCuota

Parámetros			
	monto	float	Es el monto del préstamo. Por ejemplo solicito USD 5000 al banco
	interesAnual	Int	Es la tasa de interés anual del préstamo. Por ejemplo el interés que me cobra el banco es el 16% anual
	numeroMeses	int	Es el plazo al cual me dan el préstamo. Por ejemplo me prestan a 18 meses, es decir en 18 meses debo terminar de pagar
Funcionalidad			
<p>Esta función calcula el valor de la cuota mensual que tengo que pagar para cubrir el préstamo.</p> <p>Esta cuota se calcula con la siguiente fórmula</p> $\text{Cuota mensual} = (P * i) / (1 - (1 + i)^{-n})$ <p>Donde:</p> <ul style="list-style-type: none">P corresponde al monto del préstamoi corresponde al interés mensualn corresponde al número de meses del préstamo <p>En nuestro caso la función recibe como parámetro el interés anual, para calcular el interés mensual, hay que dividir el interés anual para 12 y el resultado se divide para 100, ya que el interés llega en porcentaje.</p> <p>El símbolo \wedge significa la operación potencia. Por ejemplo 3^2 significa 3 elevado a la potencia 2. En Python para la potencia se usa **, por ejemplo 3^{**2} significa 3 elevado a la 2.</p> <p>Note que en la fórmula hay este fragmento: $(1+i)^{-n}$ que se lee (1+i) elevado a la potencia -n.</p> <p>Reemplazando por los parámetros que recibe la función y los símbolos de Python, ese fragmento quedaría:</p> $(1+\text{interesMensual})^{**}(-\text{numeroMeses})$ <p>Aplicar la fórmula completa y devolver el valor de la cuota.</p>			

Retorno	
	El valor de la cuota que se debe pagar cada mes, para cubrir el préstamo otorgado.
Test	
	Archivo: testCuota.py
	Resultados esperados: <pre>Para cubrir el crédito de USD 20000 a un interés anual del 16%, debe pagar 703.14 durante 36 meses Para cubrir el crédito de USD 5000 a un interés anual del 14%, debe pagar 240.06 durante 24 meses</pre>