

**A)(2 pts)** Escriba una función con el encabezamiento `def ganador(x1,y1,z1,x2,y2,z2)` que entregue un 1 si la primera terna (con los números enteros  $x_1, y_1$  y  $z_1$ ) le gana a la segunda terna (con los números enteros  $x_2, y_2, z_2$ ), que entregue un 2 si la segunda terna le gana a la primera, o entregue un 0 si las dos ternas son iguales.

Notas

-La terna (5,4,13) le gana a las siguientes ternas: a la terna (2,6,25) puesto que  $5 > 2$ ; a la terna (5,2,21) puesto que  $4 > 2$  y los primeros números son iguales; y a la terna (5,4,1) puesto que  $13 > 1$  y los dos primeros números son iguales.

-Debe escribir la receta de diseño incluyendo al menos 3 pruebas (una por cada resultado posible)

**B(4 pts)** El módulo grabado en el archivo `panamericanos.py` permite consultar los resultados finales de los últimos juegos panamericanos a través de las siguientes funciones:

contrato	ejemplo de uso	resultado	explicación
<code>pais: int -&gt; str</code>	<code>pais(21)</code>	"Chile"	nombre del país con el n° indicado (se numeran entre 1 y 41)
<code>oro: str -&gt; int</code>	<code>oro("Chile")</code>	13	cantidad de medallas de oro obtenidas por el país indicado
<code>plata: str -&gt; int</code>	<code>plata("Cuba")</code>	27	cantidad de medallas de plata del país indicado
<code>bronce: str -&gt; int</code>	<code>bronce("Peru")</code>	21	cantidad de medallas de bronce del país indicado

Escriba una función recursiva (sin receta de diseño) que use el módulo anterior para calcular y entregar el lugar o posición que ocupó Chile en el medallero final de los juegos.

Notas

-Utilice la función `ganador` (de la parte A) para determinar los países que le ganaron a Chile.

-Puede utilizar parámetros adicionales en su función, o bien escribir una función recursiva auxiliar.

**A(4 ptos)** Un número de Munchausen es un entero positivo que es igual a la suma de todos sus dígitos elevados a los mismos dígitos. Por ejemplo, 3435 es un número de Munchausen, puesto que la suma  $3^3 + 4^4 + 3^3 + 5^5$  es igual a 3435. Asimismo, 1 es también un número de Munchausen, puesto que  $1^1 = 1$ .

Escriba la función de encabezamiento `def esMunchausen(N)` que reciba un número entero positivo N y devuelva True si el número N es de Munchausen y False en caso contrario.

Notas.

-En caso que algún dígito del número sea igual a 0, se debe considerar que  $0^0 = 0$ . Por ejemplo, 438579088 también es un número de Munchausen, puesto que  $4^4 + 3^3 + 8^8 + 5^5 + 7^7 + 9^9 + 0^0 + 8^8 + 8^8 = 438579088$

-Debe escribir la receta de diseño incluyendo al menos 2 pruebas (una para True y otra para False)

-Defina y use una función auxiliar recursiva de encabezamiento `def suma(N)` que calcule la suma de las potencias sin escribir la receta de diseño para ella.

**B)(2 ptos)** Escriba una función recursiva (sin receta de diseño) de encabezamiento `def primero(X,Y)` que entregue el primer número de Munchausen que esté entre X e Y (ambos inclusive). Si no existe ninguno debe entregar -1 como resultado. Por ejemplo, `primero(3000,4000)` entregará el primer número de Munchausen que se encuentra entre 3000 y 4000 y `primero(2,10)` entregará -1 puesto que entre 2 y 10 no hay ningún número de Munchausen.