

Auxiliar 4: Repaso Control

Profesor: Patricio Nellef Inostroza Fajardin
Auxiliares: Alonso Almendras T., Felipe Avendaño A.,
Vicente Bórquez Z., Matías Godoy C.

P1.- (P2-C1 CC1002 2016) **Cuenta regresiva**

A(2.5 pts) Defina una función llamada `CtaRegresiva(N)`, que reciba un número entero positivo `N` y entregue un string con la cuenta regresiva desde `N` hasta el 0. Además, se debe tener la opción que si no se le entrega ningún argumento a la función, la cuenta regresiva partirá automáticamente desde el 10.

Ejemplo: `CtaRegresiva(5)` debe entregar "5 4 3 2 1 0",

Mientras que `CtaRegresiva()` entrega "10 9 8 7 6 5 4 3 2 1 0".

P2.- (P2-C1 CC1002 2019) **Munchausen**

A(4 pts) 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 $3^3 + 4^4 + 3^3 + 5^5 = 3435$. 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, siguiendo la siguiente receta de diseño (no es necesario que ustedes la escriban).

`suma(N): int -> int`

Dado un número, calcula la suma de todos los dígitos elevados a si mismos.

Ejemplo: `suma(15) = 26`, `suma(3435) = 3435` y `suma(0) = 0`

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.

P3.- (Evaluado) **Ahora es Dudeney**

Un número de Dudeney es un número entero que es un cubo perfecto y a la vez, la suma de sus dígitos da como resultado la raíz cúbica del número.

E.g.:

$$1 = 1^3 \wedge 1 = 1 \quad (1)$$

$$512 = 8^3 \wedge 8 = 5 + 1 + 2 \quad (2)$$

$$4913 = 17^3 \wedge 17 = 4 + 9 + 1 + 3 \quad (3)$$

Deberá escribir una función recursiva `primerabaseDudeney(X,Y)` que entregue el primer entero entre X e Y (ambos inclusive) que sea base de Dudeney. En caso de no existir base de Dudeney entre esos números, debe retornar -1.

Nota: Se entenderá como base de Dudeney a la raíz cúbica de un número de Dudeney.

Ejemplos de uso según los ejemplos dados:

- `primerabaseDudeney(1,6)` -> 1
- `primerabaseDudeney(1,8)` -> 1
- `primerabaseDudeney(2,3)` -> -1
- `primerabaseDudeney(12,17)` -> 17

a) Implemente la función recursiva `sumadigitos(X)`, que dado un número X, retorne la suma de los dígitos de X.

Ejemplos de uso:

- `sumadigitos(1)`-> 1
- `sumadigitos(512)`-> 8
- `sumadigitos(729)`-> 18

b) Implemente la función recursiva `primerabaseDudeney(X,Y)` usando `sumadigitos(X)`.