

Recursión en Python

Sección cátedra:
Integrantes
presentes:

| | | |
|--|-----------|--|
| | Grupo N°: | |
| | | |
| | | |
| | | |
| | | |
| | | |

Pregunta 1

Como la etiqueta de polvos Royal o la definición de recursividad, dibujen o escriban algo que represente recursividad. ¡Sean creativos!

Pregunta 2

Usando como base `Recursivo.py`, cree una función `Fibonacci(n)`, que calcule el n -ésimo número en la sucesión de Fibonacci, la cual se define por:

$$Fibonacci(n) = \begin{cases} 0, & \text{si } n = 0 \\ 1, & \text{si } n = 1 \\ Fibonacci(n - 1) + Fibonacci(n - 2), & \text{si } n > 1 \end{cases}$$

*Pista: Utilice **dos llamados recursivos** y **dos condiciones de borde**

Pregunta 3

Construye un programa que, **de forma recursiva**, determine el máximo común divisor o MCD (el mayor entero positivo que divide a ambos números de forma exacta) de dos números mediante el algoritmo de Euclides y muestre el resultado por pantalla.

El algoritmo de Euclides es el siguiente: dados dos enteros positivos m y n tales que $m > n$, hacer:

- Obtener el resto (r) de m / n .
- Si $r = 0$, el MCD es n .
- En caso contrario, el MCD para m y n es igual al MCD entre n y r .

Pista: La función recursiva tiene dos parámetros formales.

Pregunta 4

Compara ejecuciones de distintos números positivos de `iterativo(n)` y `recursivo(n)`, e indica cuál de las funciones alcanza su cota máxima primero (entrega errores en vez de resultados) y aproximadamente alrededor de qué valor se produce esta situación.