Programación Declarativa

Ingeniería Informática
Cuarto curso. Primer cuatrimestre

Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2024 – 2025

Práctica número 3.- Iteración, recursión y funciones usadas como parámetros o devueltas como resultados

Operaciones con números

1. **Conjetura de Collatz**
   - Considérese la siguiente función

$$f(n) = \begin{cases} n/2, & si\ n\ es\ par \\ 3n+1, & si\ n\ es\ impar \end{cases}$$

   - Y la sucesión numérica para un número "n"

$$a_i = \begin{cases} n, & si\ i = 0 \\ f(a_{i-1}), & si\ i > 0 \end{cases}$$

   - La conjetura de Lothar Collatz afirma que la sucesión numérica siempre se repite indefinidamente cuando alcanza los términos 4, 2, 1, independientemente del valor de "n".
   - Por ejemplo
     - Para *n* = *1*, se genera la siguiente sucesión numérica
       - 1, 4, 2, 1.
     - Para *n* = *2*, se genera la siguiente sucesión numérica
       - 2, 1, 4, 2, 1.
     - Para *n* = *3*, se genera la siguiente sucesión numérica
       - 3, 10, 5, 16, 8, 4, 2, 1.
     - Para *n* = *4*, se genera la siguiente sucesión numérica
       - 4, 2, 1.
     - Para *n* = *5*, se genera la siguiente sucesión numérica
       - 5, 16, 8, 4, 2, 1.
     - Para *n* = *6*, se genera la siguiente sucesión numérica
       - 6, 3, 10, 5, 16, 8, 4, 2, 1.
   - Codifica **dos funciones**, una <u>iterativa</u> y otra <u>recursiva</u>, que muestren la sucesión numérica de la conjetura de Collatz para un número "n" que se pasará como parámetro.

2. **Número primo**
   - Un número es **primo** si no tiene divisores propios <u>menores o iguales</u> que su raíz cuadrada.

- Codifica un predicado iterativo, denominado **primoIterativo?**, para comprobar si un número es primo o no.
- Codifica un predicado recursivo, denominado **primoRecursivo?**, para comprobar si un número es primo o no.

3. **Números afortunados de Euler**
   - Un número natural "n" es un *número afortunado de Euler* si son primos todos los números de la forma $k^2 - k - n$, donde $1 \leq k < n$.
   - Solamente existen seis números afortunados de Euler: 2, 3, 5, 11, 17 y 41.
   - Codifica una función iterativa que permita generar todos los números primos usando el polinomio de Euler $k^2 - k + n$, donde "n" es un número afortunado de Euler.

Sucesiones numéricas y límites

4. **Número e**
   - Considera el término general de una sucesión numérica que converge al número **e**: 2.718281…

$$a_n = (1 + 1/n)^n$$

   - Codifica las siguientes funciones:
     - **terminoNumeroE**
       - Calcula el término n-ésimo de la sucesión numérica.
       - Recibe como parámetro el valor de ***n*.**
     - **limiteSucesionNumeroE**
       - Se debe codificar una función iterativa que permita calcular el límite de la sucesión numérica que converge al número *e*.
       - La función debe recibir como argumento la **cota de error**, que permitirá terminar la función cuando dos elementos consecutivos de la sucesión disten menos que dicha cota de error: $|a_{n+1} - a_n| < cota$

5. **El número áureo**
   - El número áureo se define como

$$\varphi = \frac{1+\sqrt{5}}{2} = 1{,}61803398\ 9.....$$

   - Codifica una función **recursiva** denominada "**sumaAureo**" que permita calcular el número áureo usando la siguiente suma infinita:

$$\varphi = \sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+......}}}}}}$$

     - La función recibirá como parámetro el número de sumandos.
       - *(suma-aureo 0)*
         0

- *(suma-aureo 1)*
  1
- *(suma-aureo 2)*
  1.4142135623730951
- *(suma-aureo 10)*
  1.6180165422314876
- *(suma-aureo 100)*
  1.618033988749895

- Codifica una versión **iterativa** de la función.

Funciones pasadas como argumentos

6. **Fracciones continuas**
   - Una fracción continua infinita es una expresión de la forma:

$$f = \cfrac{N_1}{D_1 + \cfrac{N_2}{D_2 + \cfrac{N_3}{D_3 + \cfrac{N_4}{D_4 + \cdots}}}}$$

   - Codifica una función **iterativa**, denominada "**fracción-continua**", que permita calcular la fracción continua hasta el término k.
     - La función debe recibir tres argumentos: ***(fracción-continua N D k)***
       - **N**: función de un argumento que calcula el valor de N_k
       - **D**: función de un argumento que calcula el valor de D_k
       - **k**: número de términos de la fracción continua
   - Comprueba que la siguiente llamada a la función permite obtener una aproximación a 1/φ = 0.6180339887498948

$$\frac{1}{\varphi} = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}}}$$

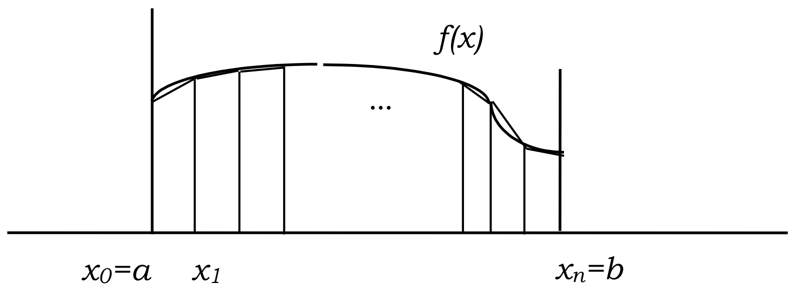
```
(fracción-continua
        (lambda (x) 1.0)  ;; función que calcula el valor de Nk
        (lambda (x) 1.0)  ;; función que calcula el valor de Dk
        k
)
```

   - Codifica una versión **recursiva** de la función "**fracción-continua**" y comprueba su funcionamiento con la llamada que calcula la aproximación a 1/φ.
     - Nota: se recomienda usar una función auxiliar local.

7. **Límite de cualquier sucesión numérica convergente**
   - Codifica una función iterativa denominada **"limiteIterativa"** que permita calcular una aproximación al límite de cualquier sucesión numérica

convergente.
- La función debe recibir como argumentos a:
  - Una **función** que represente el término general de la sucesión numérica convergente.
  - La **cota de error**, que permitirá terminar la función cuando dos elementos consecutivos de la sucesión disten menos que dicha cota de error.
- ¿Cómo se llamaría a la función "**límiteIterativa**" si se desea calcular el límite de la sucesión numérica cuyo término general es $a_n = (1 + 1/n)^n$ con una cota de error de 0.001?

8. Integral definida usando el método de los trapecios



- Codifica una función **iterativa**, denominada **integral**, que
  - reciba cuatro parámetros:
    - Los dos extremos de un intervalo: *a* y *b*
    - Una función que sea positiva en el intervalo *[a,b]: f*
    - Un número: *n*
  - y devuelva la aproximación a la integral definida según el **método de los trapecios**.

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \left(\frac{f(x_i) + f(x_{i-1})}{2} \right) * h$$

donde $h = (b - a) / n$ y $x_i = a + i * h$

  - ¿Cómo se llamaría a la función **integral** para calcular el área de la función
    - $f(x) = 1/x$ definida en el intervalo [1,2]?

9. Suma de series convergentes basadas en una cota de error

- Codifica una función **iterativa** que permita calcular **la suma de cualquier serie** numérica convergente teniendo en cuenta una **cota de error.**

$$serie = \sum_{\substack{n=inicial \\ n=n+siguiente(n)}} f(n)$$

- La función recibirá como parámetros
    - Una **función** que represente el **término general** de la serie: *f*
    - El índice del primer término: *inicial*
    - Una **función** que permita pasar al **siguiente** término de la serie: *siguiente*
    - Una **cota de error** de forma que la suma de la serie finalizará cuando el valor absoluto del término actual que se vaya a sumar sea menor que dicha cota de error: $|f(n)| < cota$
- Codifica una versión **recursiva** de la función anterior.
- **Utiliza las funciones anteriore**s para comprobar que la siguiente serie numérica permite calcular una aproximación al número *e:* 2.71828182…

$$e = \sum_{\mathrm{n}=0}^{\infty} \frac{1}{\mathrm{n}!}$$

Función "devuelta" como resultado

10. Codifica una función denominada ***suavizar*** que
- reciba como parámetros a una función ***f*** y una pequeña cantidad positiva **dx**
- y devuelva como resultado la **función suavizada** que calcularía la siguiente expresión:

$$\frac{f(x-dx)+f(x)+f(x+dx)}{3}$$

- ¿Cómo se invocaría la **función suavizada** si se desea aplicar a la función ***sqrt*** y al número 2?