Programación Declarativa

Ingeniería Informática
Cuarto curso. Primer cuatrimestre

Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2021 - 2022

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

Operaciones con números

1. **Suma digital de un número natural**
   - La suma digital de un número natural "n" es el resultado de sumar sus cifras.
     - Ejemplo: N = 39875
       - Suma digital: 3 + 9 + 8 + 7 + 5 = 32
   - Codifica tres versiones de una función que permita calcular la **suma digital** de un número natural.
     - La primera versión será **iterativa.**
     - La segunda versión será **recursiva**.
     - La tercera versión será **recursiva de cola**.

2. **Raíz digital de un número natural**
   a. Codifica una función **iterativa** que permita calcular la **raíz digital** de un número natural
      - La raíz digital de un número natural "n" es el dígito que resulta al sumar sus dígitos, volviendo a sumar repetidamente los resultados de esa suma y de las siguientes hasta que la suma sea un número de un dígito comprendido entre 1 y 9, ambos inclusive.
      - Por ejemplo
        - n = 39875
        - 3 + 9 + 8 + 7 + 5 = 32
        - 3 + 2 = 5
        - La raíz digital de n = 39875 es 5.

   b. Codifica una segunda versión de la función raíz digital utilizando la siguiente definición

$$raíz\ digital\ (n) = \begin{cases} n\ modulo\ 9 & si\ (n\ módulo\ 9)\ \neq 0 \\ 9 & si\ (n\ módulo\ 9) = 0 \end{cases}$$

3. **Persistencia aditiva de un número**
   - Si se considera el proceso de calcular la raíz digital de un número natural, la **persistencia aditiva** es el **número de veces que se han tenido que sumar** repetidamente los resultados hasta obtener un número de una cifra.
     - Por ejemplo
       - La persistencia aditiva de n = 39875 es 2
       - 3 + 9 + 8 + 7 + 5 = 32
       - 3 + 2 = 5
   - Codifica una función <u>**iterativa**</u> que permita calcular la **persistencia aditiva** de un número natural.

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

   a. Codifica un predicado <u>**iterativo**</u>, denominado **primoIterativo?**, para comprobar si un número es primo o no.

   b. Codifica un predicado <u>**recursivo**</u>, denominado **primoRecursivo?**, para comprobar si un número es primo o no.

<u>Sucesiones numéricas y límites</u>

5. **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**
       - Calculará el término n-ésimo de la sucesión numérica.
       - Recibirá como parámetro el valor de *n*.
     - **limiteSucesionNumeroE**
       - Se debe codificar una función <u>iterativa</u> 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: $|a_{n+1} - a_n| < cota$

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

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

   a. 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

b. Codifica una función **recursiva**, denominada **fracción-continua-aureo,** que permite calcular el número áureo usando la siguiente fracción continua:

$$\phi = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}}}$$

- La función recibirá como parámetro el número de fracciones continuas que debe calcular.
  - *(fraccion-continua-aureo 0)*
    1.0
  - *(fraccion-continua-aureo 1)*
    2.0
  - *(fraccion-continua-aureo 2)*
    1.5
  - *(fraccion-continua-aureo 3)*
    1.6666666666666665
  - *(fraccion-continua-aureo 10)*
    1.6179775280898876
  - *(fraccion-continua-aureo 100)*
    1.618033988749895

Sucesiones numéricas y funciones pasadas como parámetros

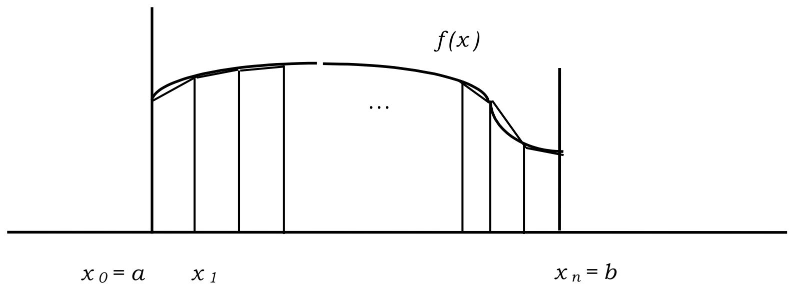
7. **Límite de cualquier sucesión numérica convergente**

a. 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.

b. ¿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?

Series numéricas y funciones pasadas como parámetros

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**
   a. 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: $|f(n)| < cota$

b. Codifica una versión **recursiva** de la función anterior.
c. **Utiliza las funciones anteriore**s para comprobar que la siguiente serie numérica permite calcular una aproximación al número *e:* 2.71828182…

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

10. **Serie de productos o "productorio"**
    a) Codifica una función **iterativa**, denominada **"productorio"**, que permita calcular cualquier **producto** definido entre dos valores:

$$productorio = \prod_{n=inicial, n=n+siguiente(n)}^{final} f(n)$$

- La función recibirá como parámetros:
    - Una **función** que represente el **término general** del productorio: ***f(n)***
    - El índice del primer término: ***inicial***
    - El índice del último término: ***final***
    - Una **función** que permita pasar al **siguiente** término de la serie: ***siguiente***

    b) Utiliza la función **"productorio"** para calcular la siguiente aproximación a $\pi/4$ propuesta por Wallis:

$$\frac{\pi}{4} = \frac{2}{3}\frac{4}{3}\frac{4}{5}\frac{6}{5}\frac{6}{7}\frac{8}{7}\frac{8}{9}\cdots$$

- Previamente, codifica una función denominada **factorWallis** que reciba como parámetro un número natural **n** y devuelva como resultado el n-ésimo factor de la sucesión de Wallis.
    - Por ejemplo:

      (factor-Wallis 1) ==> *2/3*
      (factor-Wallis 2) ==> *4/3*
      …
      (factor-Wallis 5) ==> *6 /7*
- **Observación**: la sucesión de Wallis converge "muy lentamente".

Funciones "devueltas" como resultados

11. Codifica una función denominada ***suavizar*** que reciba como parámetro a una función ***f*** con un argumento y devuelva como resultado la **función** que calcularía la siguiente expresión:

$$\frac{f(x-1)+f(x)+f(x+1)}{3}$$

- Muestra cómo se aplicaría la **función devuelta** por ***suavizar*** en los siguientes casos
    - $f(x) = x^2$, x = 0
    - f(x) = sin(x), x = 0.