Programación Declarativa

Ingeniería Informática
Cuarto curso. Primer cuatrimestre

Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2019 – 2020

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

1. **Contar cifras de un número**
   - Codifica **dos funciones**, una <u>iterativa</u> y otra <u>recursiva</u>, que permitan contar el número de cifras o dígitos de un número entero.
     - Ejemplos
       - *5 → 1 cifra*
       - *-52 → 2 cifras*
       - *502 → 3 cifras*
       - *-5021 → 4 cifras*

2. **Extraer una cifra de un número**
   - Codifica **dos funciones**, una <u>iterativa</u> y otra <u>recursiva</u>, que permitan extraer la cifra de un número entero que ocupe una determinada posición, teniendo en cuenta el sentido de derecha a izquierda: la cifra de las unidades tiene la posición 0; la cifra de las decenas, la posición 1; etc.
     - *Ejemplos*
       - *cifra 0 de 502 → 2*
       - *cifra 1 de 502 → 0*
       - *cifra 2 de 502 → 5*
       - *Si la posición no existe, se devolverá el valor -1*
       - *cifra 3 de 502 → -1*
       - *cifra -1 de 502 → -1*

3. **Suma de dígitos de un número**
   - Codifica **dos funciones**, una <u>iterativa</u> y otra <u>recursiva</u>, que sumen los dígitos de un número:
     - Ejemplos:
       - *502 → 5 + 0 + 2 = 7*
       - *7412 → 7 + 4 + 1 + 2 = 14*

4. **Reducción de un número a una sola cifra o dígito**
   - Codifica **dos funciones**, una <u>iterativa</u> y otra <u>recursiva</u>, que permitan reducir un número a una cifra entre 0 y 9
     - Ejemplos:

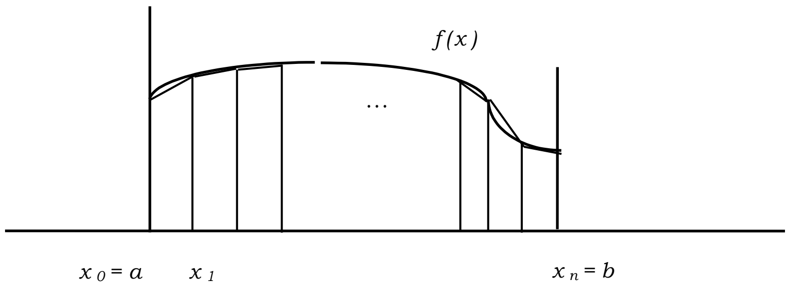
- *0 → 0*
- *7412 → 7 + 4 + 1 + 2 = 14 → 1 + 4 = 5*
- *19875 → 1 + 9 + 8 + 7 + 5 = 30 → 3 + 0 = 3*

5. **Número primo**
   - Un número es **primo** si no tiene divisores propios menores o iguales que su raíz cuadrada.
     a. Codifica un predicado iterativo, denominado **primoIterativo?**, para comprobar si un número es primo o no.
     b. Codifica un predicado recursivo, denominado **primoRecursivo?**, para comprobar si un número es primo o no.

6. **Integral definida**
   a. 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 $x_i = a + i * h$



   b. ¿Cómo se llamaría a la función **integral** para calcular el área de la función
      - $f(x) = 3x^2 + 1$ definida en el intervalo [0,3]?
   c. Utiliza las funciones explicadas en clase, basadas en el número de términos, que permiten sumar cualquier serie (versiones iterativa y recursiva) para comprobar que el resultado obtenido en el apartado "b" es correcto.

7. **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 anteriores para comprobar que la siguiente serie numérica propuesta por **Leibniz** permite calcular una aproximación a *π/4:*

$$\frac{\pi}{4} = \sum_{n=1}^{\infty} f(n) = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots$$

8. **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 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: $|a_{n+1} - a_n| < cota$

9. **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?

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

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

- El número áureo también se puede calcular mediante la siguiente suma infinita

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

   a. Codifica una función <u>iterativa</u> denominada **"sumaAureoIterativo"** que permita calcular el número áureo usando la suma anterior. La función recibirá como parámetro el número de sumandos.
   b. Codifica una función <u>recursiva</u> denominada **"sumaAureoRecursivo"** que permita calcular el número áureo usando la suma anterior. La función recibirá como parámetro el número de sumandos.

11. **Fracción continua**
- Codifica una función **iterativa** que permita calcular una aproximación a π usando la siguiente fracción continua:

$$\pi = \cfrac{4}{1 + \cfrac{1}{3 + \cfrac{2^2}{5 + \cfrac{3^2}{7 + \cdots}}}}$$

- La función recibirá como parámetro el número de fracciones continuas que debe calcular.

12. **Serie de productos o "productorio"**
- **Wallis** propuso utilizar la siguiente serie para calcular una aproximación a *π/4:*

$$\frac{\pi}{4} = \frac{2 \times 4 \times 4 \times 6 \times 6 \times 8 \times 8 \times \dots}{3 \times 3 \times 5 \times 5 \times 7 \times 7 \ v \ 9 \times \dots}$$

   a. 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*

   b. Escribe una función <u>iterativa</u> denominada **WallisIterativa** que reciba como parámetro un número natural que indicará cuántos factores se han de multiplicar.

c. Escribe función recursiva de cola denominada **WallisRecursiva** que reciba como parámetro una cota de error, de forma que la función terminará su ejecución cuando se verifique la siguiente desigualdad:

1 - cota < factor < 1 + cota

- **Observación**: la sucesión de Wallis converge "muy lentamente".

Función que devuelve otra función

13. Codifica una función denominada ***incrementoFuncional*** que reciba una función ***f*** como parámetro y devuelva como resultado la función que calcularía la siguiente expresión

$$\frac{f(x+1) - 2f(x) + f(x-1)}{4}$$

- ¿Cómo se invocaría la función ***incrementoFuncional***? Pon un ejemplo.