Programación Declarativa

Ingeniería Informática
Cuarto curso. Primer cuatrimestre

Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2022 – 2023

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

<u>Operaciones con números</u>

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

2. **Extraer una cifra de un número entero**
   - Codifica una función que permita <u>extraer</u> la cifra de un número entero que ocupe una determinada <u>posición</u>, 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. **Número capicúa**
   - Codifica un predicado, denominado "**capicua?**", que permita comprobar si un número natural es capicúa o no.
   - Ejemplos
     - *(capicua? 101)→ #t*
     - *(capicua? 1012)→ #f*
     - *(capicua? 1001)→ #t*
     - *(capicua? 1010)→ #f*

4. **Número primo**
   - Un número es **primo** si no tiene divisores propios menores o iguales 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.

Sucesiones numéricas y límites

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

6. **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

7. **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.

$$\cfrac{N_1}{D_1 + \cfrac{N_2}{\ddots + \cfrac{N_k}{D_k}}}$$

  - 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

```
(fracción-continua
        (lambda (i) 1.0)
        (lambda (i) 1.0)
        k
)
```

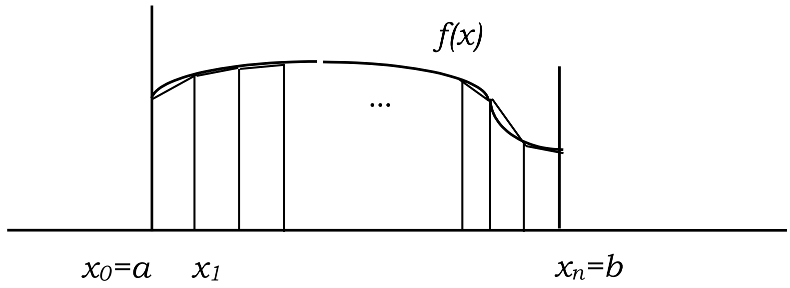
- Codifica una versión **recursiva** de la función "**fracción-continua**" y comprueba su funcionamiento con la llamada que calcula la aproximación a 1/φ.
  - Se recomienda usar una función auxiliar local.

8. **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?

9. **Integral definida usando el método de los trapecios**



- Codifica una función **<u>iterativa</u>**, 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]?

10. **Suma de series convergentes basadas en una cota de error**

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

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

  - 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 anteriores** para comprobar que la siguiente serie numérica permite calcular una aproximación al número *e:* 2.71828182…

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

Función "devuelta" como resultado

11. 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?