



Programación Declarativa

Ingeniería Informática
Cuarto curso. Primer cuatrimestre.



Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2023 - 2024

Práctica número 4.- Tipos compuestos de datos y funciones con argumentos obligatorios y opcionales

VECTORES Y MATRICES

- Codifica una función que calcule el **módulo** de un vector:
 - Si $\vec{v} = (x_1, x_2, \dots, x_n)$ entonces su módulo se define como
 - $|\vec{v}| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$
 - Importante**
 - La función se podrá aplicar a vectores de cualquier longitud.
 - Ejemplos**
 - (modulo #(1 0 2))* \rightarrow 2,236067
 - (modulo #(1 0 2 0 3))* \rightarrow 3,741657
- Codifica una función iterativa que permita calcular la **media aritmética** de un vector $\vec{v} = (x_1, x_2, \dots, x_n)$
 - Media aritmética**
$$\frac{x_1 + x_2 + \dots + x_n}{n}$$
 - Importante**
 - La función se podrá aplicar a vectores de cualquier longitud.
 - Ejemplo**
 - (mediaAritmetica #(1. 2. 3.))* \rightarrow 2.0
 - (mediaAritmetica #(1. 2. 3. 4. 5.))* \rightarrow 3.0
- Codifica una función, denominada “**extraer-columna**”, que permita extraer la columna de una matriz de dos dimensiones
 - Ejemplos**
 - 1. 2. 3. 2.
 - 4. 5. 6. 4.
 - 7. 8. 9. 7.
 - (extraer-columna #(#(1. 2. 3. 2.) #(4. 5. 6. 4.) #(7. 8. 9. 7.)) 1)* \rightarrow *#(2. 5. 8.)*
- Codifica una función, denominada “**maxiMin**”, que reciba una matriz (no necesariamente cuadrada) y devuelva el máximo de los valores mínimos de las columnas de la matriz.

- **Ejemplo**
 1. 2. 3. 2.
 4. 5. 6. 4.
 7. 8. 9. 7.
 - (*maxiMin* #(#(1. 2. 3. 2.) #(4. 5. 6. 4.) #(7. 8. 9. 7.))) → 3.

5. Determinante de una matriz de dimensión 3 x 3 y área del triángulo.
- a) Codifica una función que permita calcular el **determinante** de una matriz de dimensión 3 x 3:
- $$\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = x_1y_2z_3 + x_2y_3z_1 + x_3y_1z_2 - x_1y_3z_2 - x_2y_1z_3 - x_3y_2z_1$$
- b) Utiliza la función anterior para calcular el **área de un triángulo** definido por sus vértices: $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$ y $P_3=(x_3,y_3)$
- $\text{área}(P_1,P_2,P_3) = \text{valor_absoluto} \left(\frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \right)$

LISTAS Y PARÁMETROS OBLIGATORIOS U OPCIONALES

6. Codifica una función, denominada *primosMenores*, que reciba un número natural y genere una lista con los números **primos menores o iguales** que dicho número.
- **Ejemplos**
 - (*primosMenores* 9) → (2 3 5 7)
 - (*primosMenores* 11) → (2 3 5 7 11)
 - **Nota:**
 - Utiliza el predicado “*primo?*” codificado en la práctica nº 3.
7. Codifica una función, denominada *descomposiciónEnPrimos*, que reciba un número natural y genere una lista con su **descomposición en números primos**.
- **Ejemplos**
 - (*descomposiciónEnPrimos* 2) → (2)
 - (*descomposiciónEnPrimos* 12) → (2 2 3)
 - (*descomposiciónEnPrimos* 60) → (2 2 3 5)
8. Codifica una función, denominada *filtrarListaPrimos*, que reciba una lista de números naturales y devuelva otra lista compuesta por los números primos.
- **Ejemplos**
 - (*filtrarListaPrimos* ‘()’) → ()
 - (*filtrarListaPrimos* ‘(2 3 4 5 6)’) → (2 3 5)
 - (*filtrarListaPrimos* ‘(2 3 4 5 6 7 8 9 10)’) → (2 3 5 7)
9. Codifica una función denominada *filtrarPrimos* que reciba un número variable de números naturales y devuelva una lista compuesta por los números primos.

- Ejemplos
 - $(\text{filtrarPrimos }) \rightarrow ()$
 - $(\text{filtrarPrimos } 2\ 3\ 4\ 5\ 6) \rightarrow (2\ 3\ 5)$
 - $(\text{filtrarPrimos } 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12) \rightarrow (2\ 3\ 5\ 7\ 11)$

10. Codifica una función denominada ***filtrarPrimosDelimitados*** que reciba dos números “inicial” y “final” (parámetros obligatorios) y un número variable de números naturales y devuelva una lista compuesta por los números primos delimitados por “inicial” y “final”.

- Ejemplos
 - $(\text{filtrarPrimosDelimitados } 2\ 10) \rightarrow ()$
 - $(\text{filtrarPrimosDelimitados } 2\ 10\ 3\ 4\ 5\ 6) \rightarrow (3\ 5)$
 - $(\text{filtrarPrimosDelimitados } 2\ 10\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12) \rightarrow (3\ 5\ 7)$
 - $(\text{filtrarPrimosDelimitados } 10\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12) \rightarrow ()$

11. Codifica una función recursiva, denominada ***separar***, que reciba como parámetro una lista de números y los reparta en dos listas, dependiendo de que ocupen un "lugar o posición" par o impar.

- Ejemplos
 - $(\text{separar } '()) \rightarrow (()\ ())$
 - $(\text{separar } '(2)) \rightarrow ((2)\ ())$
 - $(\text{separar } '(3\ 2)) \rightarrow ((3)\ (2))$
 - $(\text{separar } '(1\ 3\ 2)) \rightarrow ((1\ 2)\ (3))$
 - $(\text{separar } '(4\ 1\ 3\ 2)) \rightarrow ((4\ 3)\ (1\ 2))$
 - $(\text{separar } '(5\ 4\ 1\ 3\ 2)) \rightarrow ((5\ 1\ 2)\ (4\ 3))$

12. Codifica una función recursiva, denominada ***unir***, que reciba como parámetros dos listas ordenadas de números y devuelva otra lista con los números ordenados

- Ejemplos
 - $(\text{unir } '()\ '()) \rightarrow ()$
 - $(\text{unir } '(1)\ '()) \rightarrow (1)$
 - $(\text{unir } '(1)\ '(2)) \rightarrow (1\ 2)$
 - $(\text{unir } '(1\ 3)\ '(2)) \rightarrow (1\ 2\ 3)$
 - $(\text{unir } '(1\ 3)\ '(2\ 4\ 5)) \rightarrow (1\ 2\ 3\ 4\ 5)$
 - $(\text{unir } '(1\ 3\ 5)\ '(2\ 4)) \rightarrow (1\ 2\ 3\ 4\ 5)$
 - $(\text{unir } '()\ '(1\ 2\ 3\ 4\ 5)) \rightarrow (1\ 2\ 3\ 4\ 5)$

13. Método de ordenación ***mergeSort***

- Descripción
 - Datos de entrada: 5 4 1 3 2
 - División
 - ✓ Primera: 5 1 2 | 4 3
 - ✓ Segunda: 5 2 | 1 || 4 | 3
 - ✓ Tercera: 5 | 2 || 1 ||| 4 | 3
 - Fusión:
 - ✓ Primera: 2 5 | 1 || 3 4

✓ Segunda: 1 2 5 | 3 4

✓ Tercera: 1 2 3 4 5

- Codifica una función que permita ordenar una **lista** de números utilizando el método *mergeSort*.
 - Ejemplo
 - ✓ (*mergeSort* '(5 4 1 3 2)) → (1 2 3 4 5)
- **Observación**
 - Utilizas las funciones “*separar*” y “*unir*” de los ejercicios anteriores.

14. Codifica una función denominada *mergeSortDatos* que permita ordenar una **cantidad variable** de números utilizando el método *mergeSort*.

- **Ejemplo**
 - (*mergeSortDatos*) → ()
 - (*mergeSortDatos* 2) → (2)
 - (*mergeSortDatos* 1 3 2) → 1 2 3)
 - (*mergeSortDatos* 5 4 1 3 2) → (1 2 3 4 5)