



Lenguajes de Inteligencia Artificial

Segundo curso. Primer cuatrimestre

Ingeniería Técnica en Informática de Gestión
e Ingeniería Técnica en Informática de Sistemas
Escuela Politécnica Superior



Universidad de Córdoba
Curso académico: 2010 - 2011

Práctica número 4: vectores, listas, tipos abstractos de datos y entrada y salida

1. Escribe una función **iterativa** que realice el producto escalar de dos vectores de longitud variable:
(producto_escalar #(1 0 2 0 1) #(1 2 3 4 5)) → 12
2. (*) Codifica las siguientes funciones
 - a. Una función **iterativa** que calcule el mínimo de un vector
(mínimo-vector #(9 4 7 2 6)) → 2
 - b. Una función **iterativa** que calcule el máximo de una matriz no necesariamente cuadrada: (mínimo-matriz #(8 4 3) #(5 1 7))) → 1
 - **Observación:**
 - Utilícese la función **mínimo-vector** del apartado anterior.
3. Codifica una función que permita escribir una matriz
 - a. Por filas
 - b. Por columnas
4. (*) Codifica una función **recursiva**, denominada **suprimir**, que reciba como parámetro una lista de objetos l, que puede tener sublistas, y un elemento x y dé como resultado otra lista en la que no aparezca x.
(suprimir '(a b d c (a b a) (d (e g) f) b) 'a) → (b d c (b) (d (e g) f) b)
5. Codifica una función **recursiva**, denominada **eliminar-sublistas**, que reciba como parámetro una lista l compuesta por átomos y sublistas y dé como resultado otra lista en la que no aparezcan las sublistas de l.
(eliminar-sublistas '((a b d) a (c a) b c d (e))) → (a b c d)
6. (*) Codifica una función **recursiva**, denominada **diferencia**, que reciba como parámetros dos listas y dé como resultado otra lista compuesta por los elementos de la primera lista que no pertenecen a la segunda lista.
(diferencia '(a b c d) '(c d e f g)) → (a b)
7. (*) Codifica una función **recursiva**, denominada **diferencia-simétrica**, que reciba como parámetros dos listas y dé como resultado otra lista compuesta por los elementos que sólo pertenecen a uno y a sólo una de las listas.
(diferencia-simétrica '(a b c d) '(c d e f g)) → (a b e f g)
8. (*) Codifica una función **recursiva** denominada **media-geométrica** que reciba "n" argumentos opcionales: a_1, a_2, \dots, a_n y que calcule su media geométrica

$$\text{media - geométrica}(a_1, a_2, \dots, a_{1n}) = \sqrt[n]{a_1 \times a_2 \times \dots \times a_n}$$

(media-geométrica 2 3 4) → 2,884499...

9. (*) Codifica una función **recursiva** que reciba una lista de números naturales y devuelva otra lista compuesta sólo por los números primos.

(*filtrar-lista-primos* '(2 4 5 15 17 33)) → (2 5 17)

- **Observación:**

- Utilícese un predicado auxiliar, denominado **primo?**, codificado en la práctica nº 3.

10. (*) Codifica una función recursiva, denominada **filtrar_primos**, que reciba un **número variable** de números y dé como resultado una lista compuesta sólo por los números primos:

(*filtrar-primos* 2 3 4 5 6 7 8 9) → (2 3 5 7)

11. (*) Codifica un predicado denominado **datos-extremos?** que reciba dos argumentos obligatorios denominados “inferior” y “superior” y un número **variable** de argumentos y compruebe si cada uno de los datos variables está comprendido entre el extremo inferior y el superior.

(*datos-extremos?* 1 10 7 5 9) → #t

(*datos-extremos?* 1 10 9 19) → #f

12. (*) Codifica las funciones de **creación, acceso, consulta y modificación** del tipo abstracto de datos **erasmus**, que posee los campos **nombre, titulación y país** para representar a un estudiante con una beca de Erasmus.

- a. Versión con vectores
- b. Versión con listas simples
- c. Versión con listas de asociación

13. (*) Utiliza las funciones del tipos abstracto **erasmus** para codificar las siguientes funciones del tipo abstracto “**lista-erasmus**”:

- a. **crear-lista-erasmus-vacia**: crea una lista vacía de estudiantes eramus.
- b. **lista-erasmus-vacia?**: comprueba si una lista de estudiantes eramus está vacía.
- c. **insertar-erasmus**: inserta un estudiante erasmus en una lista - erasmus
- d. **borrar-erasmus**: borra un estudiante erasmus de una lista - erasmus
- e. **primer-erasmus**: recibe una lista de erasmus y devuelve el primer elemento
- f. **resto-de-eramus**: recibe una lista de erasmus y devuelve todos sus elementos excepto el primero.
- g. **mostrar-erasmus**: recibe una lista de estudiantes y muestra por pantalla todos sus datos.
- h. **mostrar-erasmus-país**: recibe una lista de estudiantes y un país y muestra los estudiantes que disfrutan de una beca Funci?n Funci?n de Erasmus en dicho país.
- i. **grabar-lista-erasmus**: graba en un fichero los datos de los estudiantes erasmus.
- j. **cargar-fichero-lista-erasmus**: lee los datos de los estudiantes erasmus desde un fichero y los introduce en una lista de erasmus.

- **Observación**

- Estas funciones sólo podrán acceder a los campos de erasmus a través de las funciones del tipo abstracto de datos de erasmus.